AI in Fundamental Physics

Sven Krippendorf, 20.11.2025
Durham ATM

Menu

- Inverse problems in fundamental physics: flux vacua in the string landscape
- NN dynamics $\leftrightarrow$ Field theory dynamics
- Automatising Fundamental Physics research using LLMs

“Give me string models that realise an EFT at low-energies consistent with all experiments and observations.”

The full-fledged problem is hard but some problems we can address.
e.g.: scale of supersymmetry breaking $|W_0|$ and string coupling g_s

Flux vacua as a toy model for BSM models

Broader physics motivation

Flux vacua as a toy model for BSM models

Broader physics motivation

"Give me string models that realise $|W_0| = 100$."

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

"Give me string models that realise $|W_0| = 100$."

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

"Give me string models that realise $|W_0| = 100$."

"Give me BSM models that solve the Hubble tension."

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

“Give me string models that realise $|W_0| = 100$.”

“Give me BSM models that solve the Hubble tension.”

“What is the conditional density of flux vectors $P(\mathbf{x} \,|\, W_0)$?”

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

"Give me string models that realise $|W_0| = 100$."

"Give me BSM models that solve the Hubble tension."

"What is the conditional density of flux vectors $P(\mathbf{x} \,|\, W_0)$?"

"What is the number of flux vacua with $|W_0| = 100$ and $N_{\text{flux}} < 10$?"

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

“Give me string models that realise $|W_0| = 100$.”

“Give me BSM models that solve the Hubble tension.”

Definite and probabilistic answers

“What is the conditional density of flux vectors $P(\mathbf{x}\,|\,W_0)$?”

“What is the number of flux vacua with $|W_0| = 100$ and $N_{\text{flux}} < 10$?”

Flux vacua as a toy model for BSM models

Broader physics motivation

Model builder

"Give me string models that realise $|W_0| = 100$."

"Give me BSM models that solve the Hubble tension."

Definite and probabilistic answers

"What is the conditional density of flux vectors $P(\mathbf{x} \,|\, W_0)$?"

"What is the number of flux vacua with $|W_0| = 100$ and $N_{\text{flux}} < 10$?"

"What is the probability of primordial GWs at high frequencies from a consistent theory of quantum gravity?"

Methodwise: Why is this a good question to automatize?

Motivation for using flux vacua

Related work on generative models in mathematical physics:
Halverson, Long; SK, Erbin; Seong

- Well-defined model building task (finite search)
- Phenomenological relevance
- Non-linear problem, i.e. only special solutions around (e.g. $D_i W = 0$ many solutions not available in closed form.
- Benchmark against other numerical and human approaches.

Methodwise: Why is this a good question to automatize?

Motivation for using flux vacua

Related work on generative models in mathematical physics:
Halverson, Long; SK, Erbin; Seong

- Well-defined model building task (finite search)
- Phenomenological relevance
- Non-linear problem, i.e. only special solutions around (e.g. $D_i W = 0$ many solutions not available in closed form.
- Benchmark against other numerical and human approaches.

Conditional probabilistic models are ubiquitous in AI.

Methodwise: Why is this a good question to automatize?

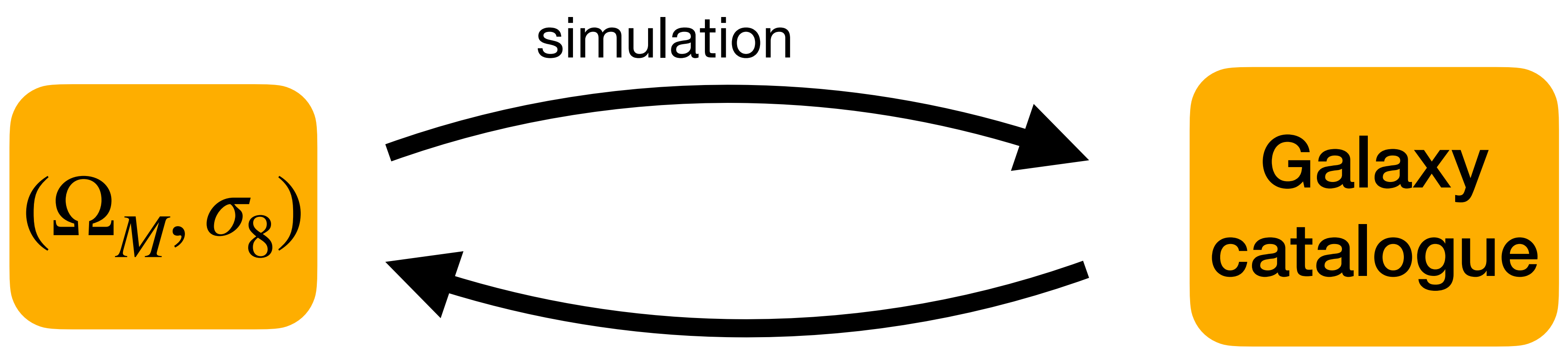
Motivation for using flux vacua

Related work on generative models in mathematical physics:
Halverson, Long; SK, Erbin; Seong

- Well-defined model building task (finite search)
- Phenomenological relevance
- Non-linear problem, i.e. only special solutions around (e.g. $D_i W = 0$ many solutions not available in closed form.
- Benchmark against other numerical and human approaches.

Conditional probabilistic models are ubiquitous in AI.

e.g. cosmological parameter inference



$P(\Omega_M, \sigma_8 \,|\, \text{obs. galaxy catalogue})$

e.g. Lehman, SK, Weller, Dolag 2411.08957

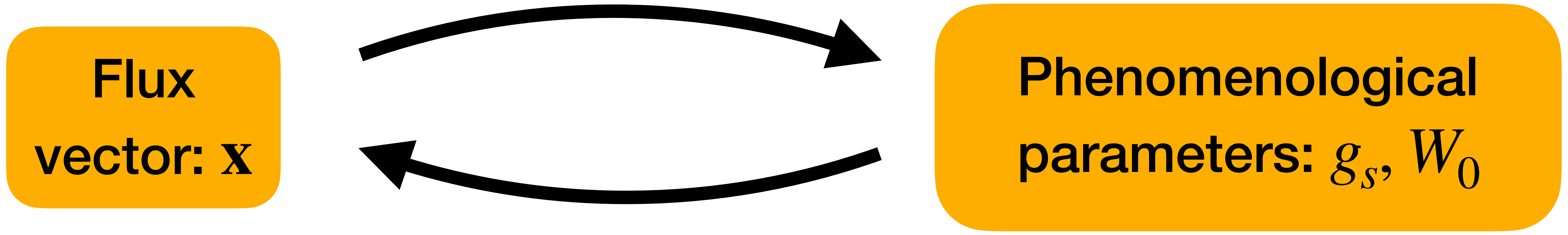
Today: the AI string model builder

A model in realistic situations for questions like: $P(\mathbf{x}\,|\,W_0)$

Solving inverse problems of Type IIB flux vacua with conditional generative models

SK, Liu: 2506.22551

see also Walden and Larfors



Today: the AI string model builder

A model in realistic situations for questions like: $P(\mathbf{x} \mid W_0)$

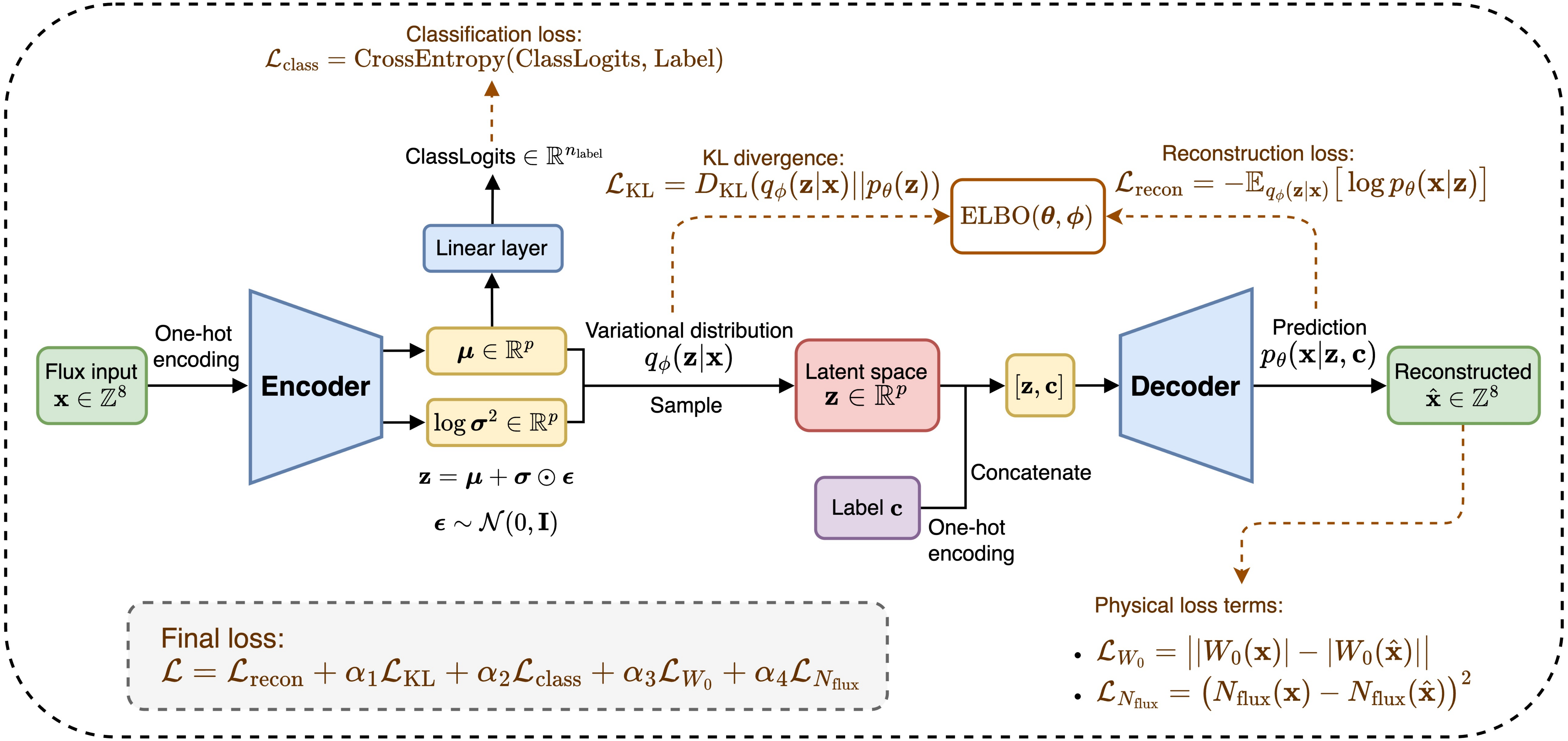
Solving inverse problems of Type IIB flux vacua with conditional generative models

SK, Liu: 2506.22551

see also Walden and Larfors

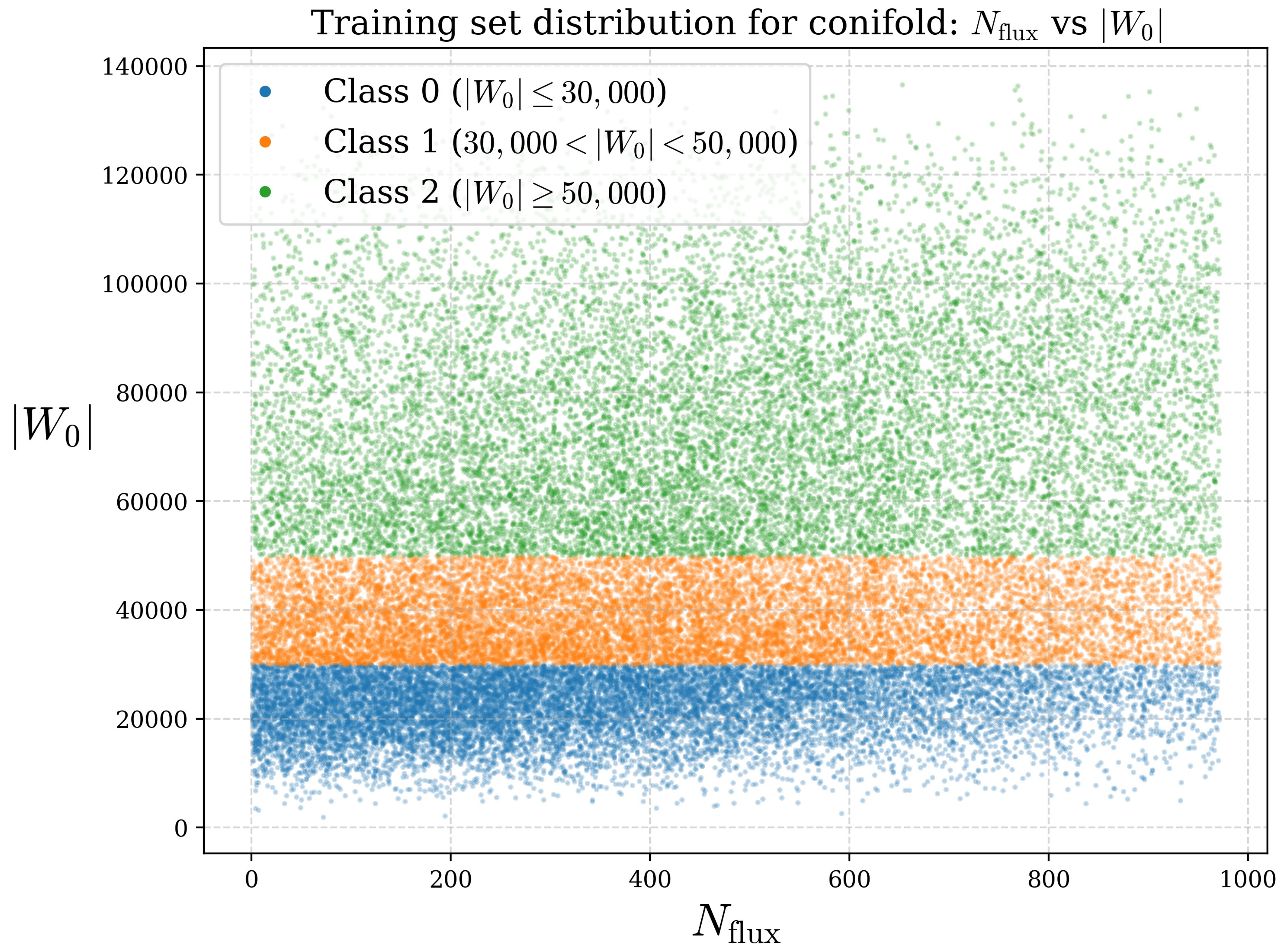
Customised Conditional Variational Autoencoder

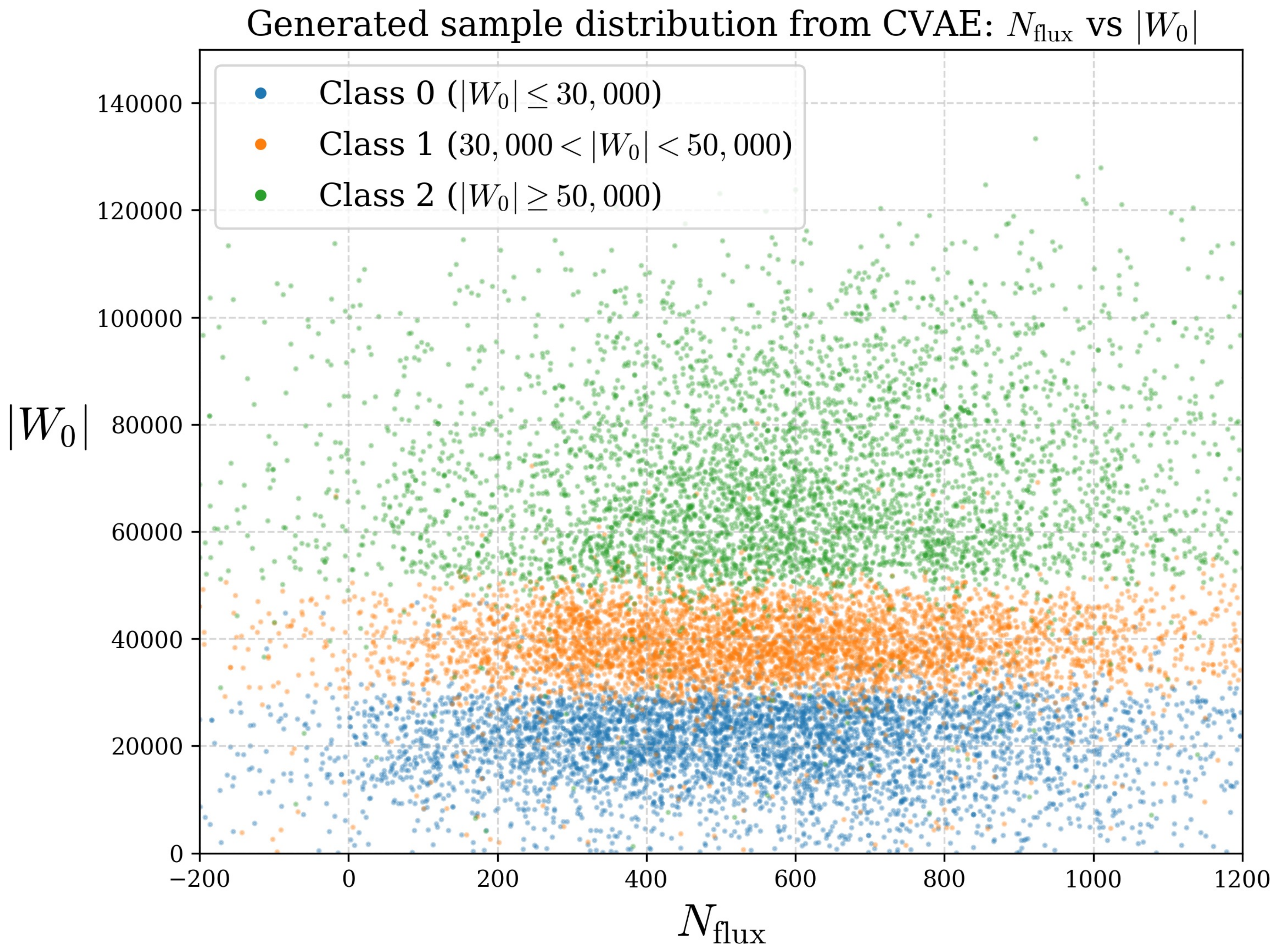
Standard losses and physics-inspired losses



CVAE samples $P(\mathbf{x}\,|\,\text{Class})$

Can you sample flux vacua with particular phenomenological properties?





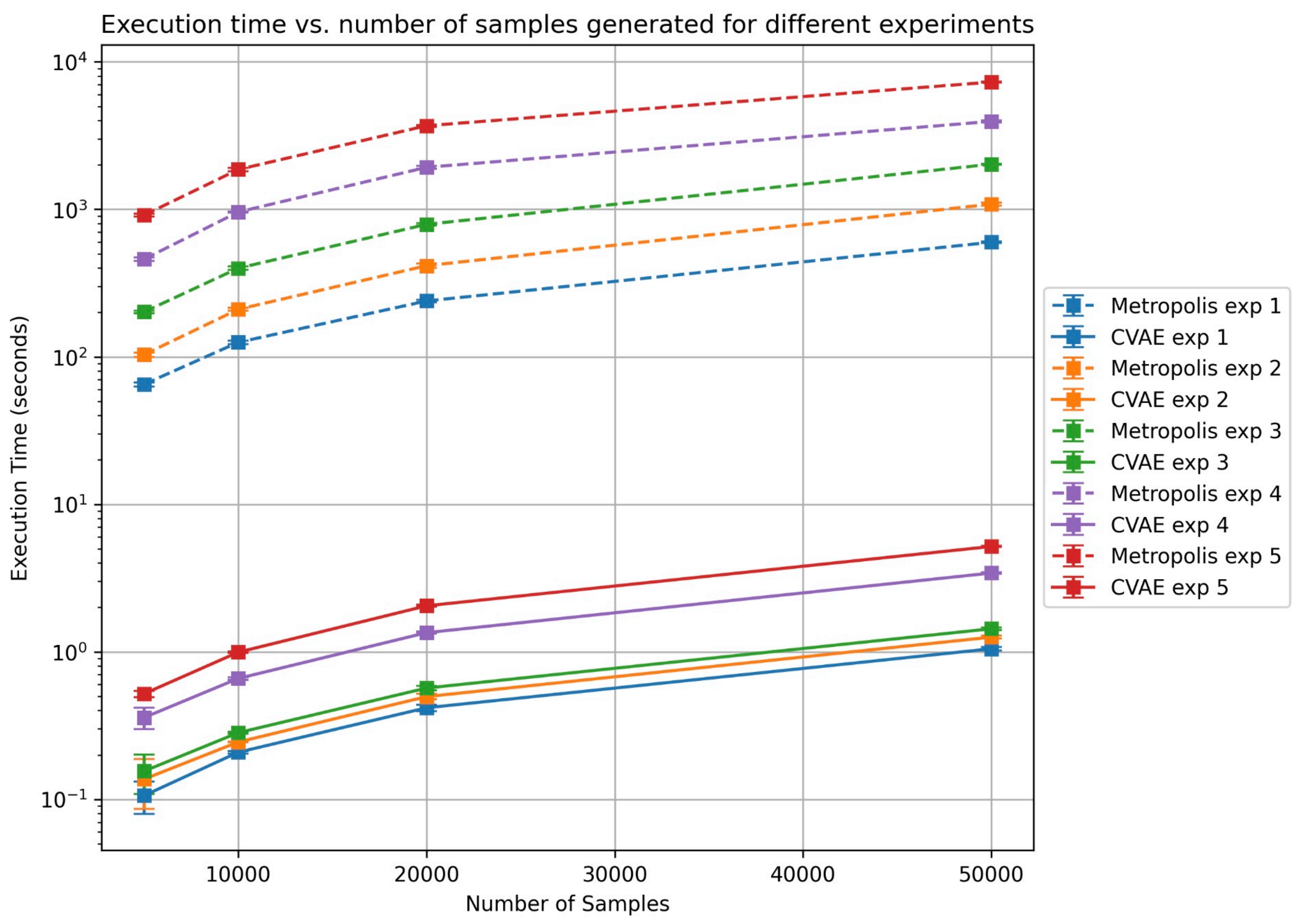
Here: conifold and torus setup as in 0411061, 2107.04039, 2111.11466, 2209.15433.

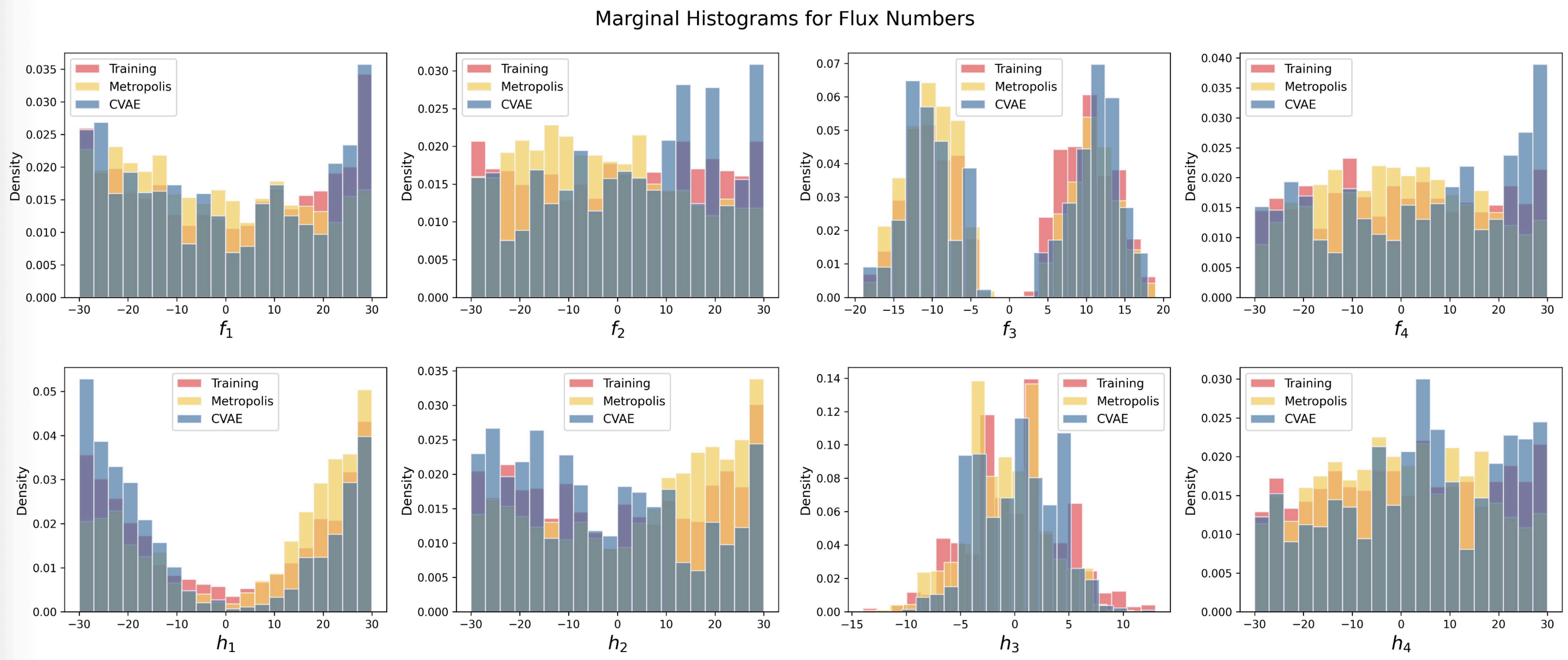
CVAE results

Learning training distribution, focusing on specific values possible

- Metropolis sampling orders of magnitudes less efficient.

Exp	$\lvert W_0 \rvert$ range	# training	# CVAE generated	# x	# distinct x
1	$40{,}000 \pm 10{,}000$	12,028	100,000	$66{,}596 \pm 181$	$50{,}837 \pm 182$
2	$40{,}000 \pm 5{,}000$	5868	100,000	59028 ± 128	39469 ± 40
3	$40{,}000 \pm 2{,}500$	3550	100,000	49147 ± 109	26599 ± 27
4	$40{,}000 \pm 1{,}000$	1843	100,000	26700 ± 42	11979 ± 83
5	$40{,}000 \pm 500$	992	100,000	13209 ± 80	5975 ± 31

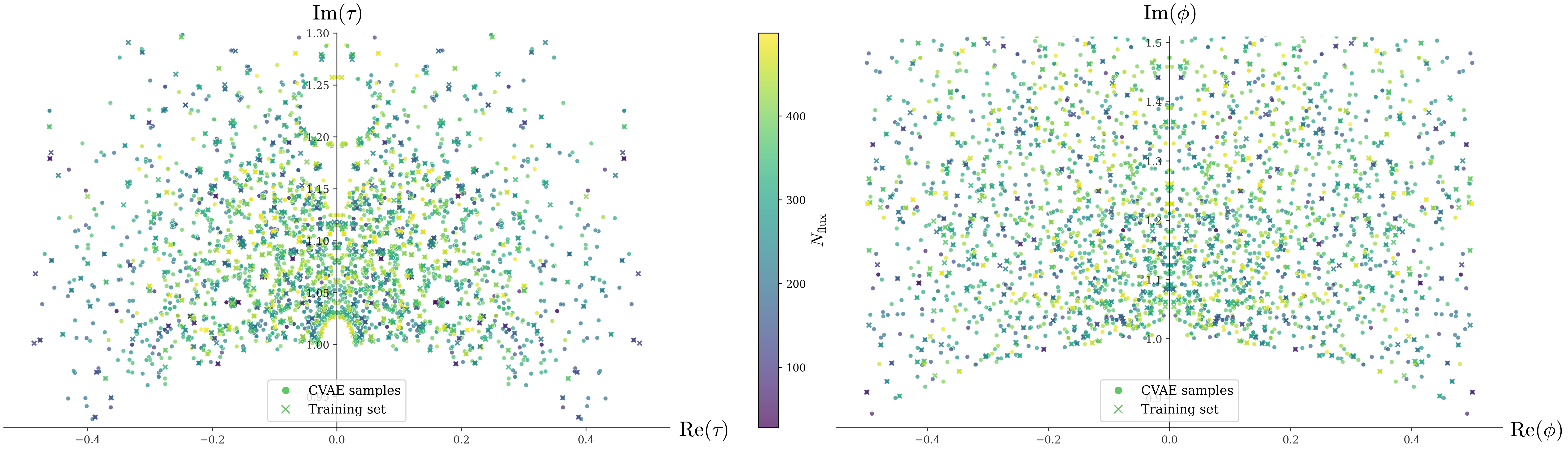




CVAE: generating new samples

Torus example: similar results

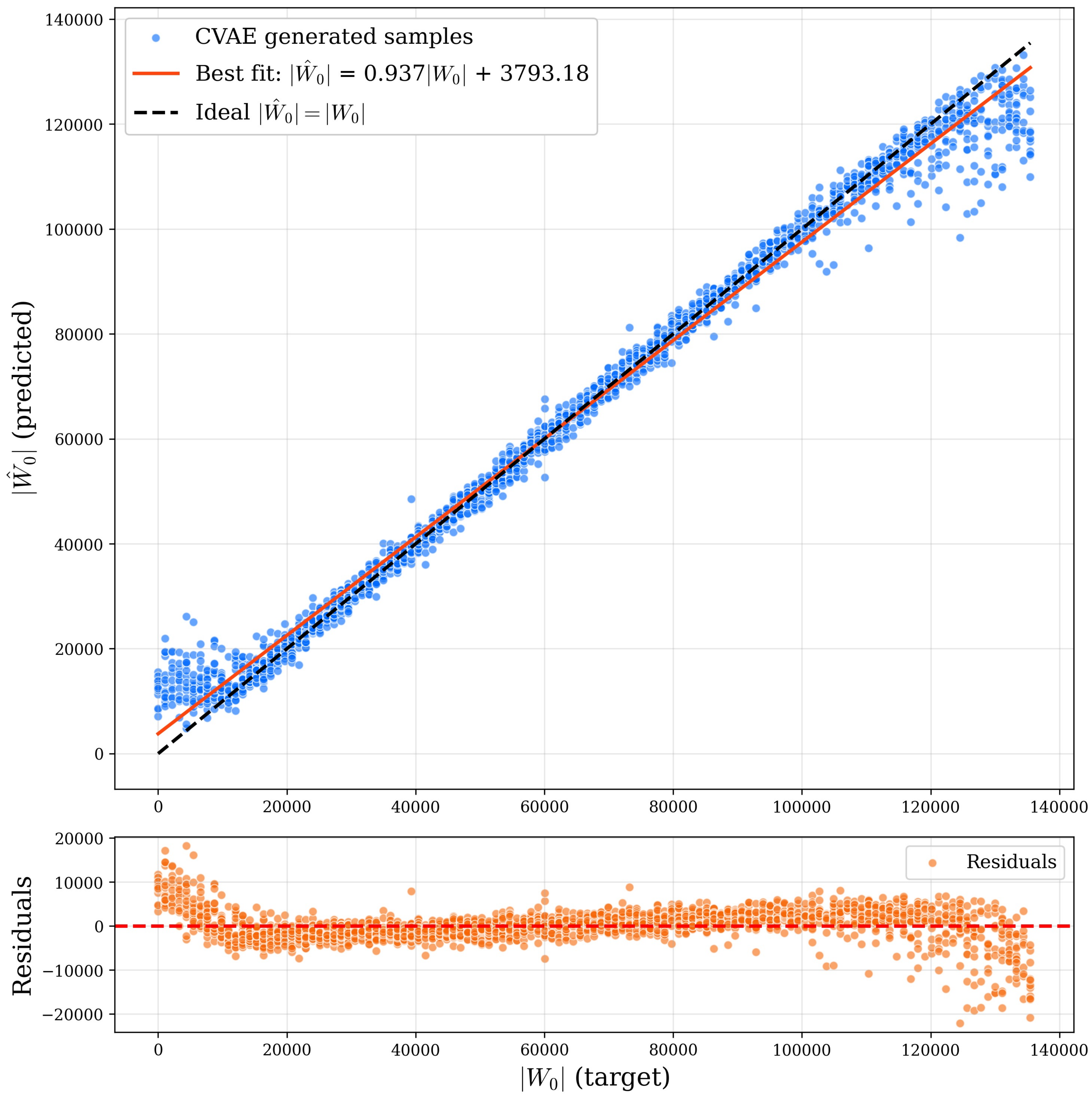
- Here, we can use CVAE to augment the training set, i.e. we can really generate viable new vacua.



CVAE: regression

Sampling $P(\mathbf{x} \,|\, |W_0|)$

- We can not just target one value but sample in a range as well (regression).
- Genetic algorithm only for targeted searches and reinforcement learning approaches not as efficient.
(cf. 2107.04039, 2111.11466, 2209.15433)
- Completeness?
- Scaling?



Completeness

"What is the number of flux vacua with $|W_0| = 100$ **and** $N_{\text{flux}} < 10$**?"**

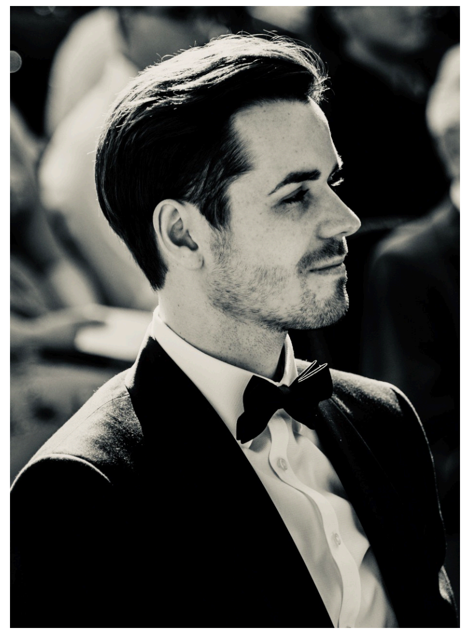
Deep observations of regions of moduli space in $\mathbb{P}_{1,1,1,6,9}$

work in collaboration with (2501.03984):

Aman Chauhan

Michele Cicoli

Anshuman Maharana



Andreas Schachner

Pelegrino Piantadosi

How do #fluxes become manageable?

Focusing on regions of moduli space

cf. Plauschinn
[here slightly stronger bounds]

- Rewriting the ISD condition $\star_6 G_3 = \mathrm{i}G_3$:

Fluxes: $f, h \in \mathbb{Z}^{2h^{2,1}+2}$

$$f = \left(s\,\Sigma \cdot \mathscr{M} + c_0 \mathbf{1}\right) \cdot h$$

$$\mathscr{M} = \begin{pmatrix} -\mathscr{I}^{-1} & \mathscr{I}^{-1}\mathscr{R} \\ \mathscr{R}\mathscr{I}^{-1} & -\mathscr{I} - \mathscr{R}\mathscr{I}^{-1}\mathscr{R} \end{pmatrix}$$

$$\mathscr{N} = \mathscr{R} + \mathrm{i}\mathscr{I}$$

Prepotential

$$\mathscr{N}_{IJ} = \overline{F}_{IJ} + 2\mathrm{i}\,\frac{\mathrm{Im}(F_{IL})X^L\,\mathrm{Im}(F_{JK})X^K}{X^M\mathrm{Im}(F_{MN})X^N}\,, \quad F_{IJ} = \partial_{X^I}\partial_{X^J}F\,.$$

- Finite region of moduli (eigenvalues of M) and tadpole constrain allowed fluxes:

$$|h|^2 \leq \frac{2N_{\mathrm{flux}}\lambda_{\mathrm{max}}}{\sqrt{3}} \qquad \frac{\sqrt{3}}{2}\,\frac{N_{\mathrm{flux}}}{\lambda_{\mathrm{max}}} \leq |f|^2 \leq \frac{\lambda_{\mathrm{max}}^2 N_{\mathrm{flux}}^2}{|h|^2} + \frac{|h|^2}{4}\,.$$

The flux vacua universe

Models meet observations

Statistical models (continuous flux approximation):

$$\mathcal{N}_{\text{stat}}(N_{\text{flux}} \leq N_{\text{max}}) = \frac{(2\pi N_{\text{max}})^6}{6!} \int_{\mathcal{M}_\tau \times \mathcal{M}_{\text{cs}}} \mathrm{d}^6 z \, \det(g) \, \rho(z)$$

$$\rho(z) = \pi^{-6} \int \mathrm{d}^2 X \, \mathrm{d}^4 Z \, \mathrm{e}^{-|X|^2 - |Z|^2} \, |X|^2 \left| \det \begin{pmatrix} \delta^{IJ} \, \overline{X} - \frac{\overline{Z}^I Z^J}{X} & F_{IJK} \, \overline{Z}^K \\ \overline{F}_{IJK} \, Z^K & \delta^{IJ} X - \frac{Z^I \, \overline{Z}^J}{\bar{X}} \end{pmatrix} \right|$$

Do deep observations of flux landscape reproduce such estimates?

Algorithmic biases in observed ensembles? Can we quantify those biases?

Douglas, Ashok, Denef

Our telescope for deep observations of flux vacua: JAXvacua

Model construction
CY orientifold data from computational tools

↓

EFT module
Auto-diff to construct SUGRA equations from prepotential

↓

Sampling module
Choice of initial guesses for moduli and fluxes

↓

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

↓

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Nutshell overview

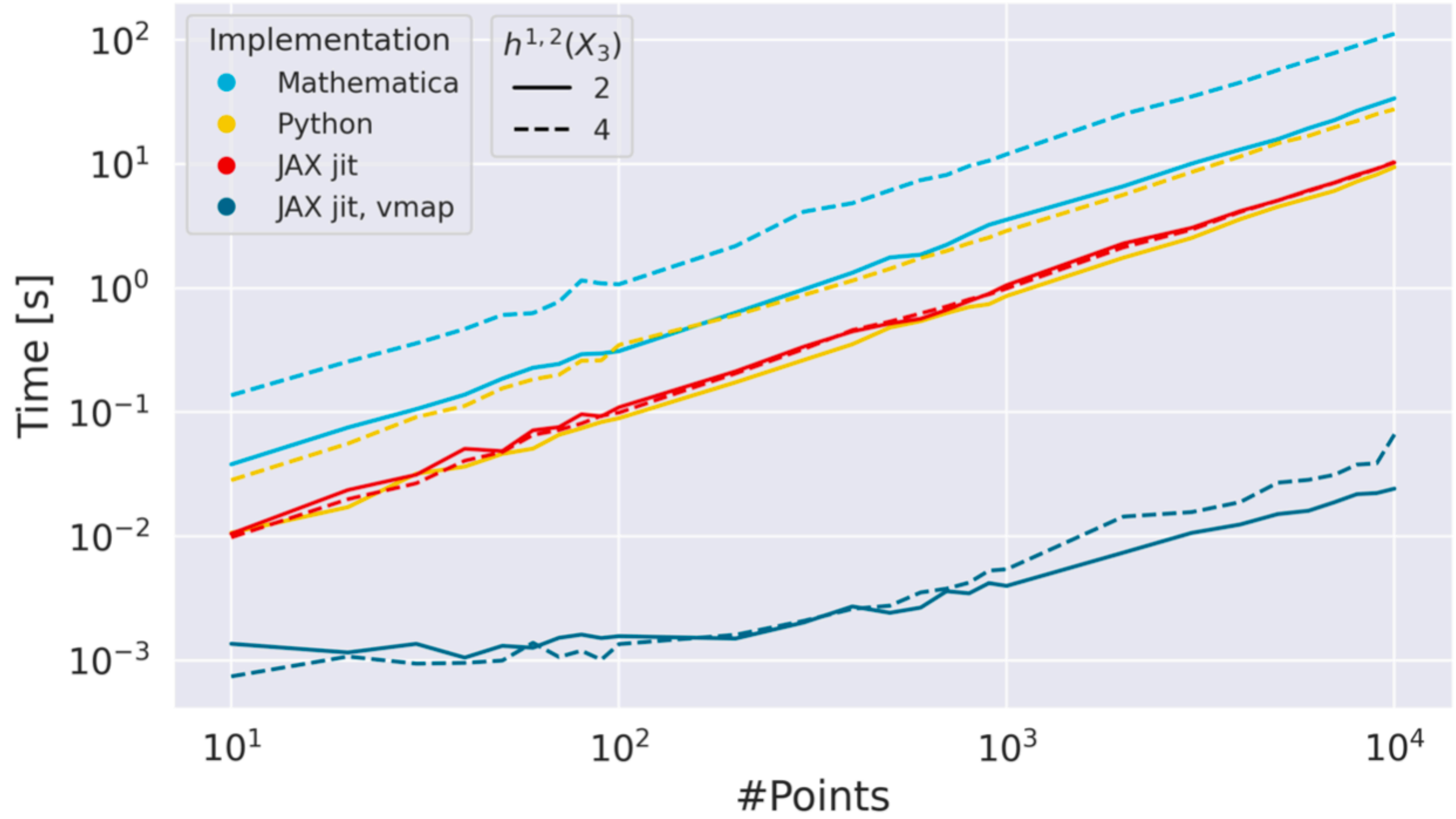
- Flexible code (i.e. re-use for different CY data) for EFT properties with JAX

CY data		Prepotential		Periods		EFT properties
$\kappa_{ijk}\,,c_2\,,GVs$	⇢	F	⇢ auto-diff	Π, K, W	⇢ auto-diff	$V\,,\partial_I\partial_J V\,,M_{3/2}\,,\,\ldots$

Only hardcoded input!

- Auto-diff: machine precision derivatives.
- JIT for compiled code, VMAP for vectorization

Timing for evaluating $D_I W$



Orders of magnitude speed improvements!

Deep explorations

- Fix tadpole and region in moduli space (fixes range for eigenvalue spectrum of matrix in ISD).
- Generate box of flux vectors for h (sample points in region of moduli space), f fixed from ISD.
- Find flux vacua using JAXvacua.
- Check equivalences, consistency (masses, LCS valid)

$$f = \left(s\,\Sigma \cdot \mathscr{M} + c_0 \mathbf{1}\right) \cdot h$$

$$|h|^2 \leq \frac{2N_{\text{flux}}\lambda_{\max}}{\sqrt{3}}$$

$$\frac{\sqrt{3}}{2}\,\frac{N_{\text{flux}}}{\lambda_{\max}} \leq |f|^2 \leq \frac{\lambda_{\max}^2 N_{\text{flux}}^2}{|h|^2} + \frac{|h|^2}{4}\,.$$

Our deep observations

Our deep observations

Four datasets

- Old friend: $\mathbb{P}_{[1,1,1,6,9]}$ symmetric locus, large complex structure.

Name	$\mathrm{Im}(z^i)$	s	$N_{\max}$	$\#h$	$\#f$	$\#(f,h)$	$\mathcal{N}_{\mathrm{vac}}$	exhaustive
A	$[2,3]$	$[\frac{\sqrt{3}}{2},20]$	34	82,082	1,849,426	5,134,862	5,140,872	✓
B	$[2,5]$	$[\frac{\sqrt{3}}{2},10]$	10	1,900	6,340	12,160	12,196	✓
C	$[1,10]$	$[\frac{\sqrt{3}}{2},50]$	34	3,652,744	21,043,832	50,652,686	50,884,086	×
D	$[2,10]$	$[\frac{\sqrt{3}}{2},10]$	50	5,909,012	45,886,900	123,075,206	123,408,240	×

coarse study e.g. 1212.4530 (Westphal et al.)

Our deep observations

Four datasets

- Old friend: $\mathbb{P}_{[1,1,1,6,9]}$ symmetric locus, large complex structure.

Name	$\text{Im}(z^i)$	s	N_{max}	$\#h$	$\#f$	$\#(f,h)$	$\mathcal{N}_{\text{vac}}$	exhaustive
A	$[2,3]$	$[\frac{\sqrt{3}}{2},20]$	34	82,082	1,849,426	5,134,862	5,140,872	✓
B	$[2,5]$	$[\frac{\sqrt{3}}{2},10]$	10	1,900	6,340	12,160	12,196	✓
C	$[1,10]$	$[\frac{\sqrt{3}}{2},50]$	34	3,652,744	21,043,832	50,652,686	50,884,086	×
D	$[2,10]$	$[\frac{\sqrt{3}}{2},10]$	50	5,909,012	45,886,900	123,075,206	123,408,240	×

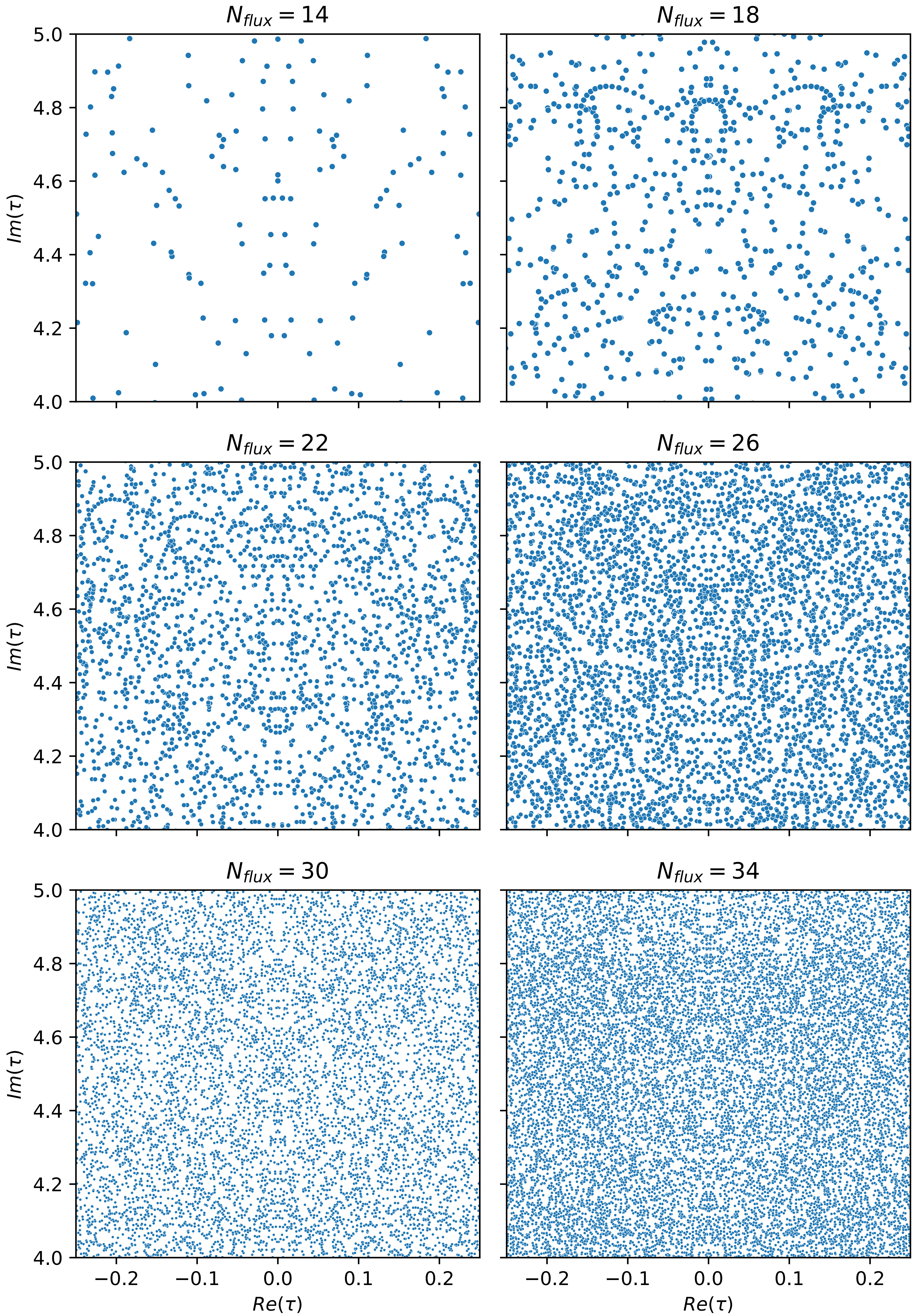
coarse study e.g. 1212.4530 (Westphal et al.)

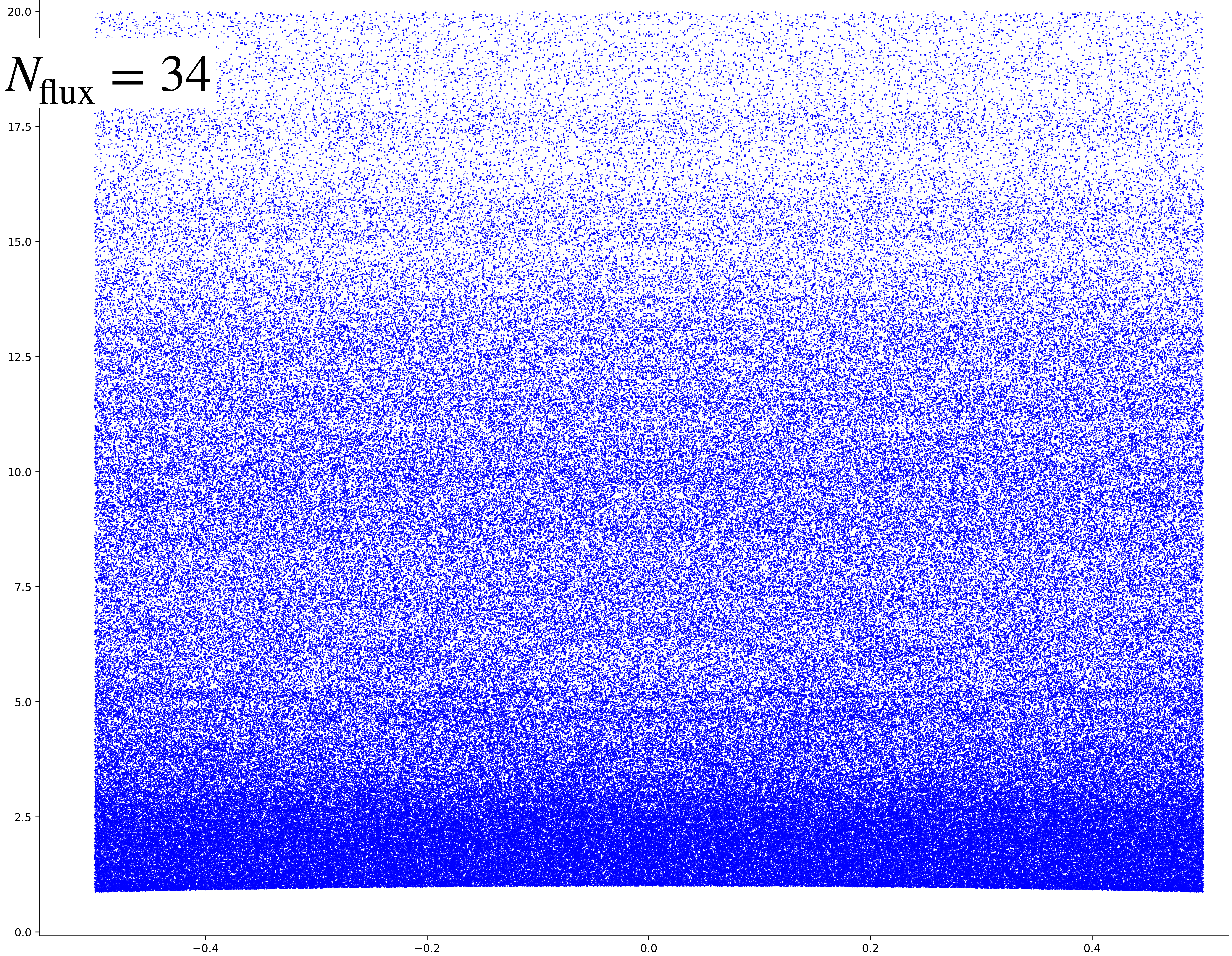
Dilaton solutions

Dataset A

- Structures in string couplings revealed when filtering individual N_{flux} values.
- Scale of the structures changes when changing N_{flux}.

Name	$\text{Im}(z^i)$	s	N_{max}	$\#h$	$\#f$	$\#(f,h)$	$\mathcal{N}_{\text{vac}}$	exhaustive
A	$[2,3]$	$[\frac{\sqrt{3}}{2}, 20]$	34	82,082	1,849,426	5,134,862	5,140,872	✓
B	$[2,5]$	$[\frac{\sqrt{3}}{2}, 10]$	10	1,900	6,340	12,160	12,196	✓
C	$[1,10]$	$[\frac{\sqrt{3}}{2}, 50]$	34	3,652,744	21,043,832	50,652,686	50,884,086	×
D	$[2,10]$	$[\frac{\sqrt{3}}{2}, 10]$	50	5,909,012	45,886,900	123,075,206	123,408,240	×



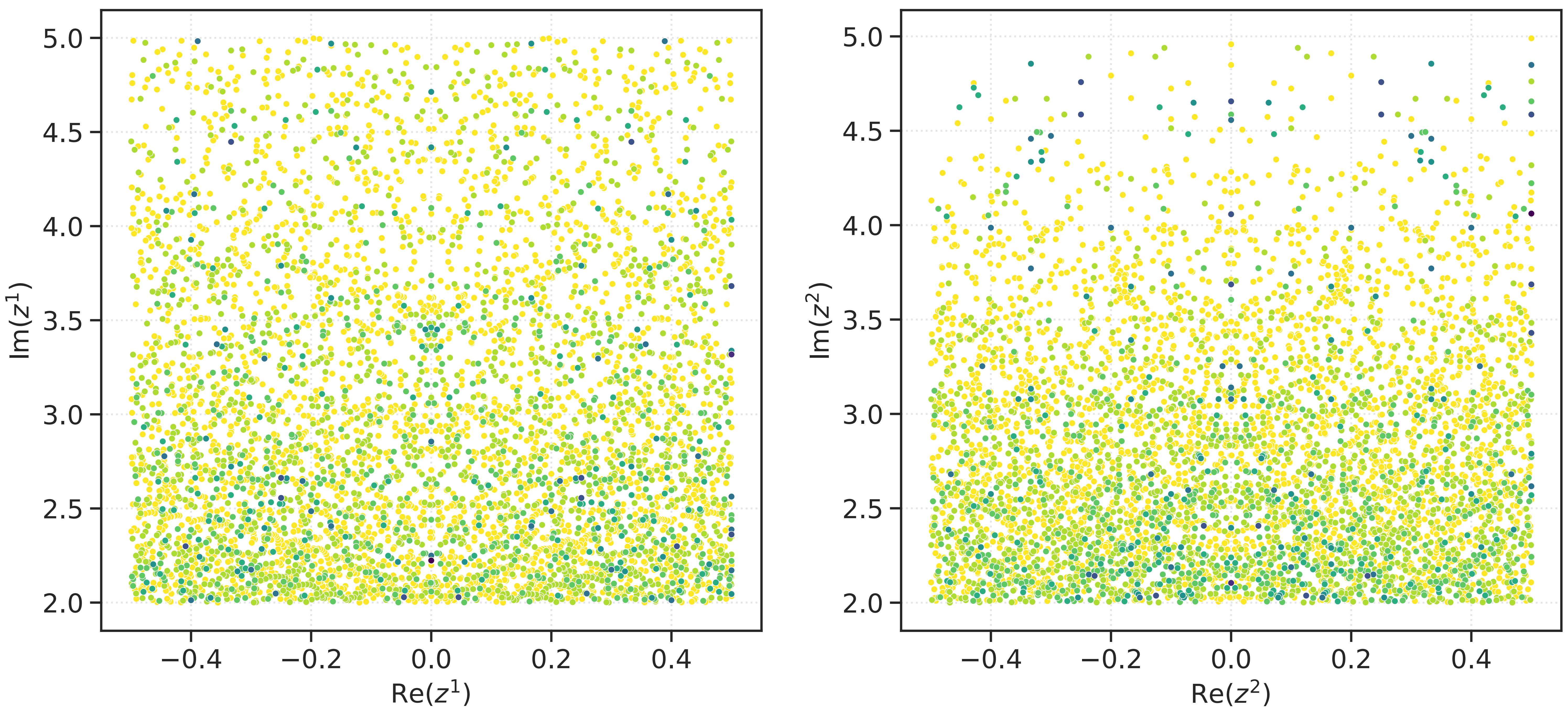


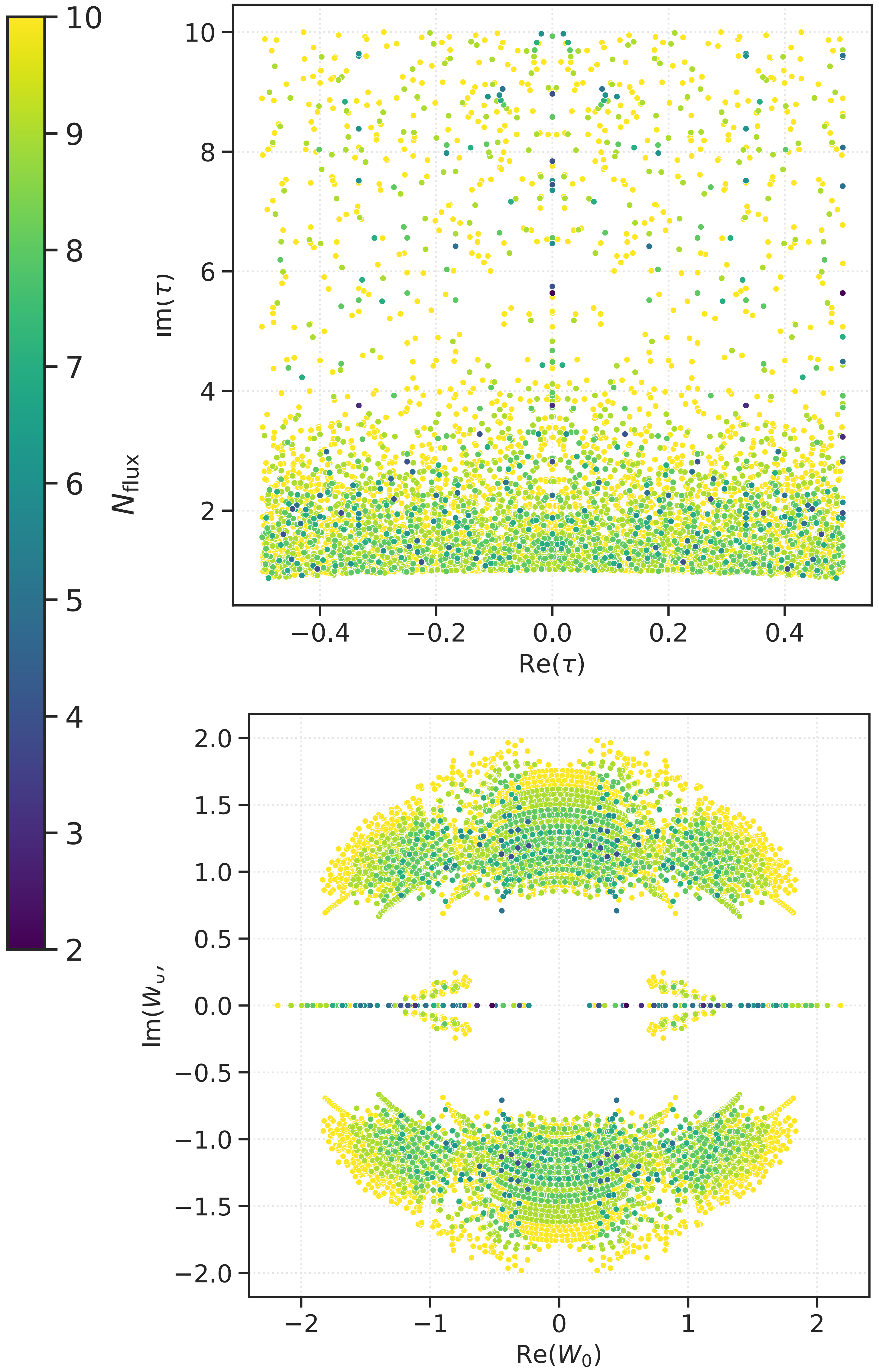
$N_{\text{flux}} = 34$
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−0.4
−0.2
0.0
0.2
0.4


$N_{\mathrm{flux}} = 34$

$$N_{\text{flux}} = 34$$

Distribution of solutions (dataset B)

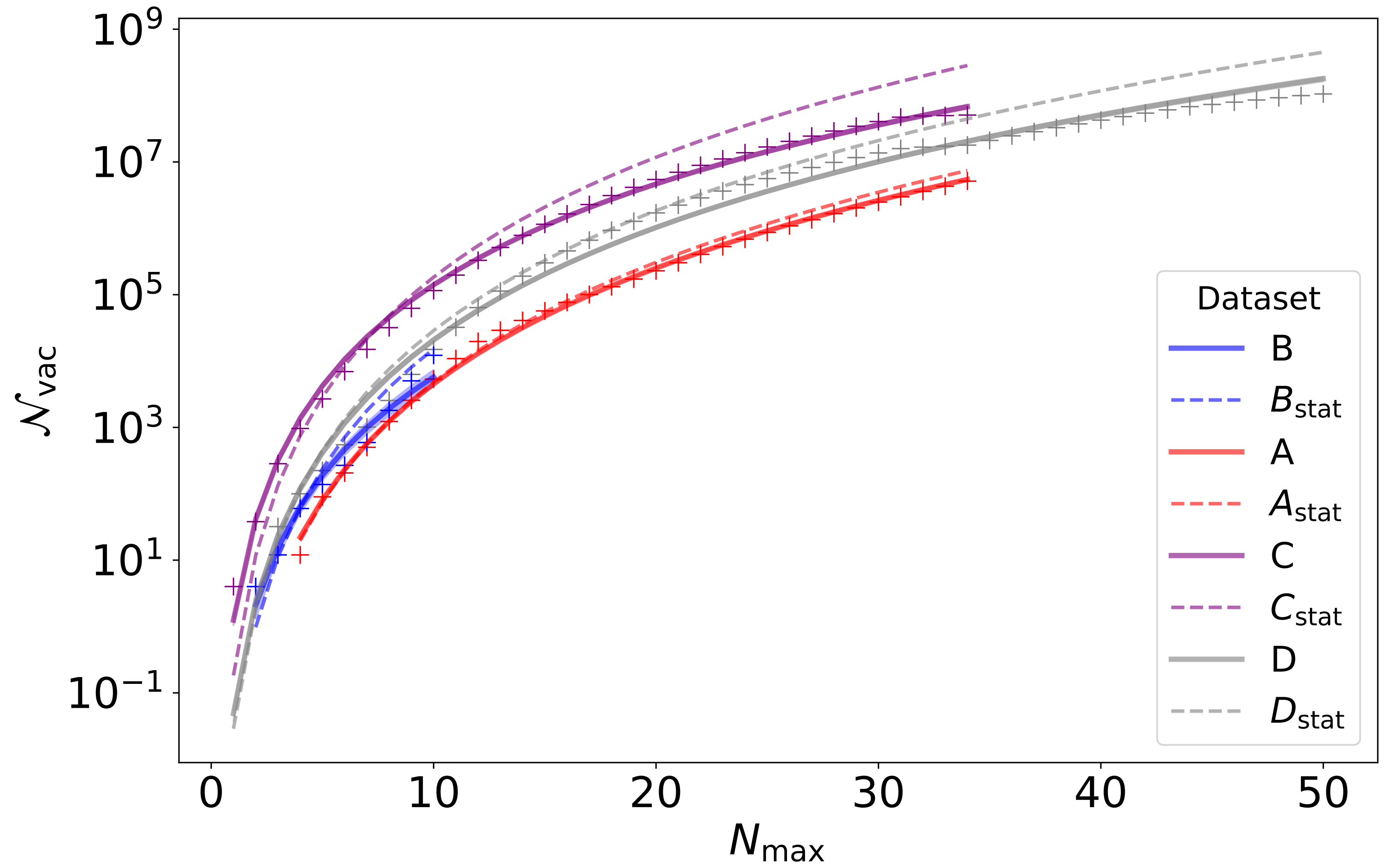




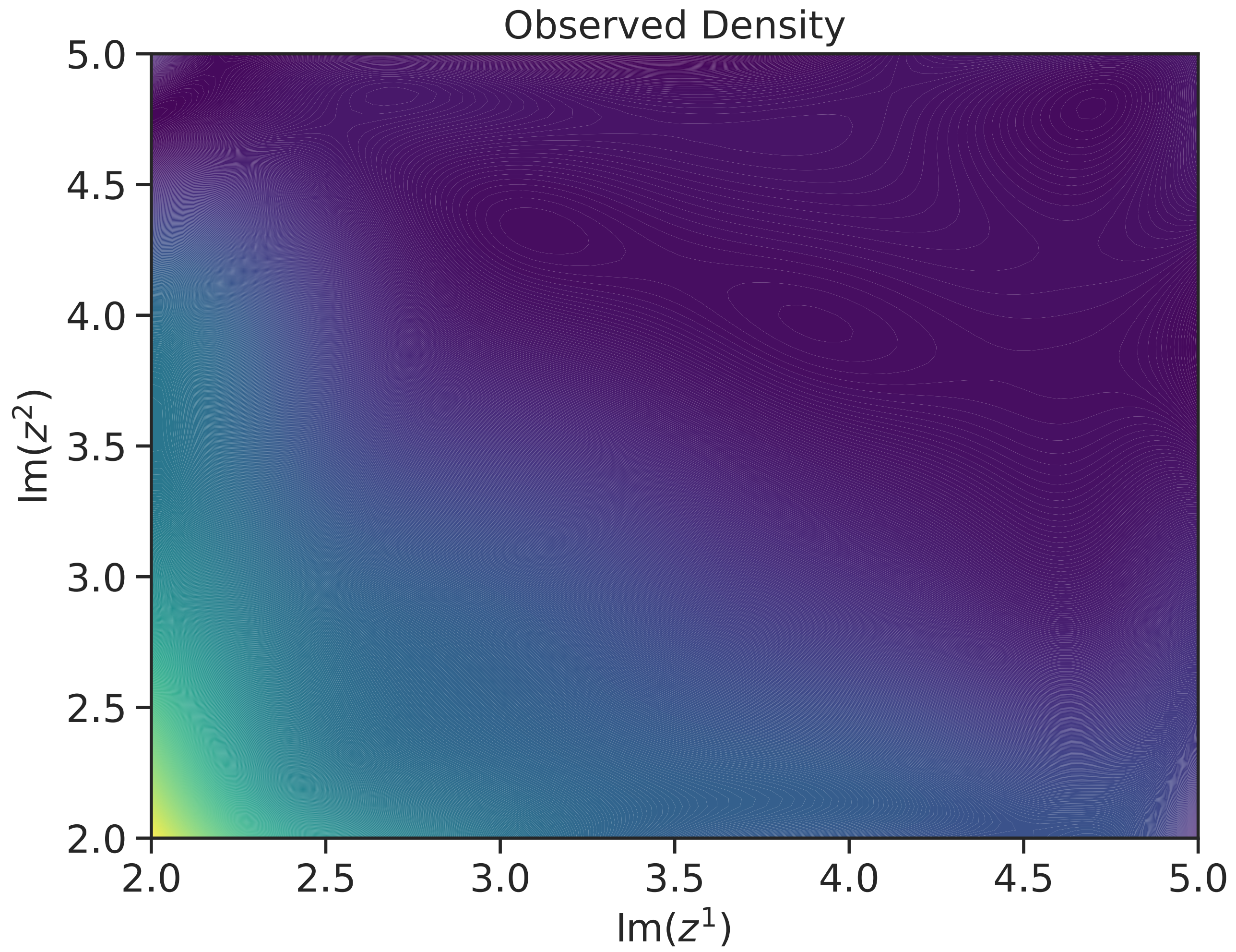
Name	Im(z^i)	s	$N_{\max}$	$\#h$	$\#f$	$\#(f,h)$	$\mathcal{N}_{\text{vac}}$	exhaustive
A	$[2,3]$	$[\frac{\sqrt{3}}{2},20]$	34	82,082	1,849,426	5,134,862	5,140,872	✓
B	$[2,5]$	$[\frac{\sqrt{3}}{2},10]$	10	1,900	6,340	12,160	12,196	✓
C	$[1,10]$	$[\frac{\sqrt{3}}{2},50]$	34	3,652,744	21,043,832	50,652,686	50,884,086	×
D	$[2,10]$	$[\frac{\sqrt{3}}{2},10]$	50	5,909,012	45,886,900	123,075,206	123,408,240	×

Local deviations from statistical expectations

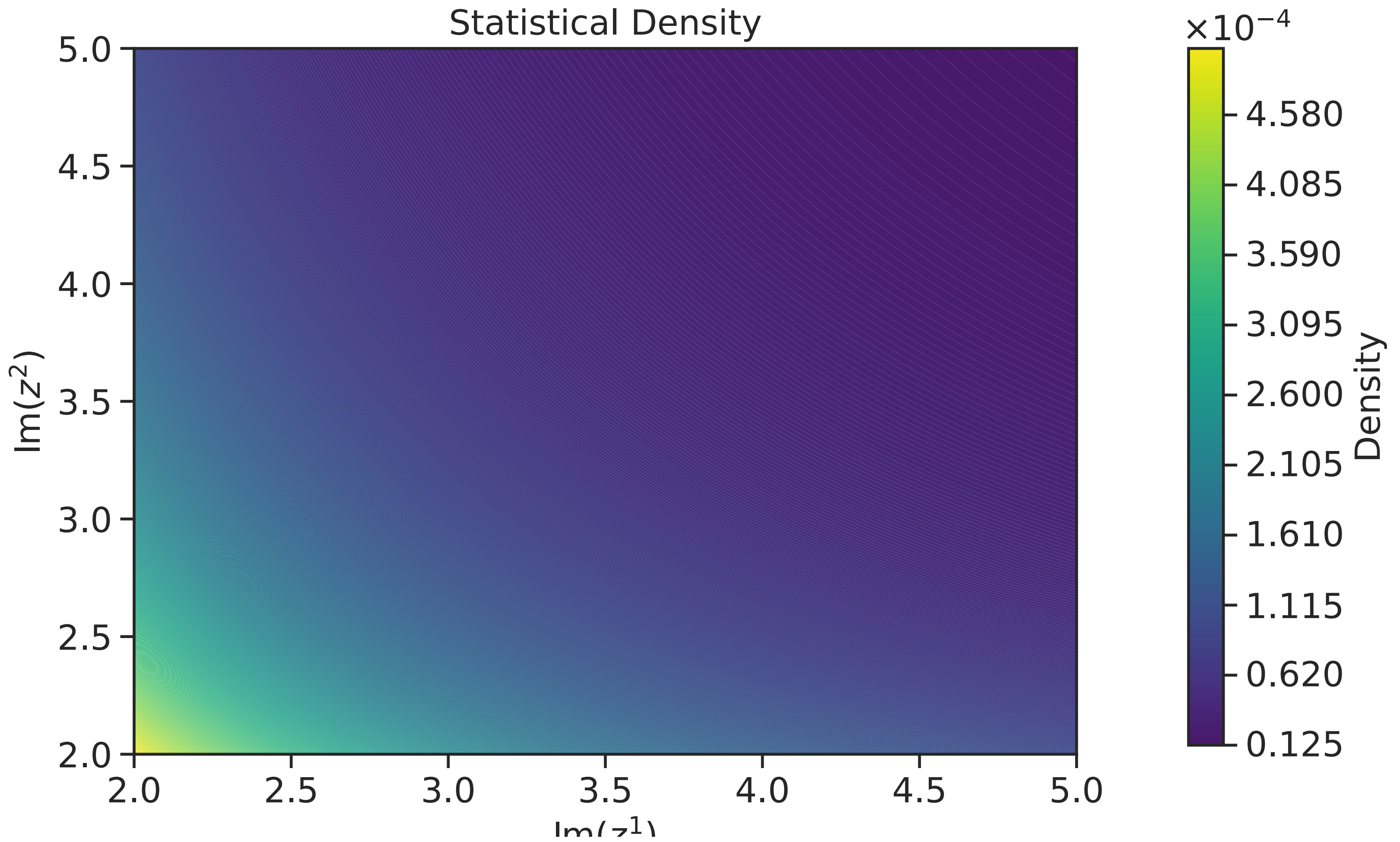
Expectations vs. observed total numbers

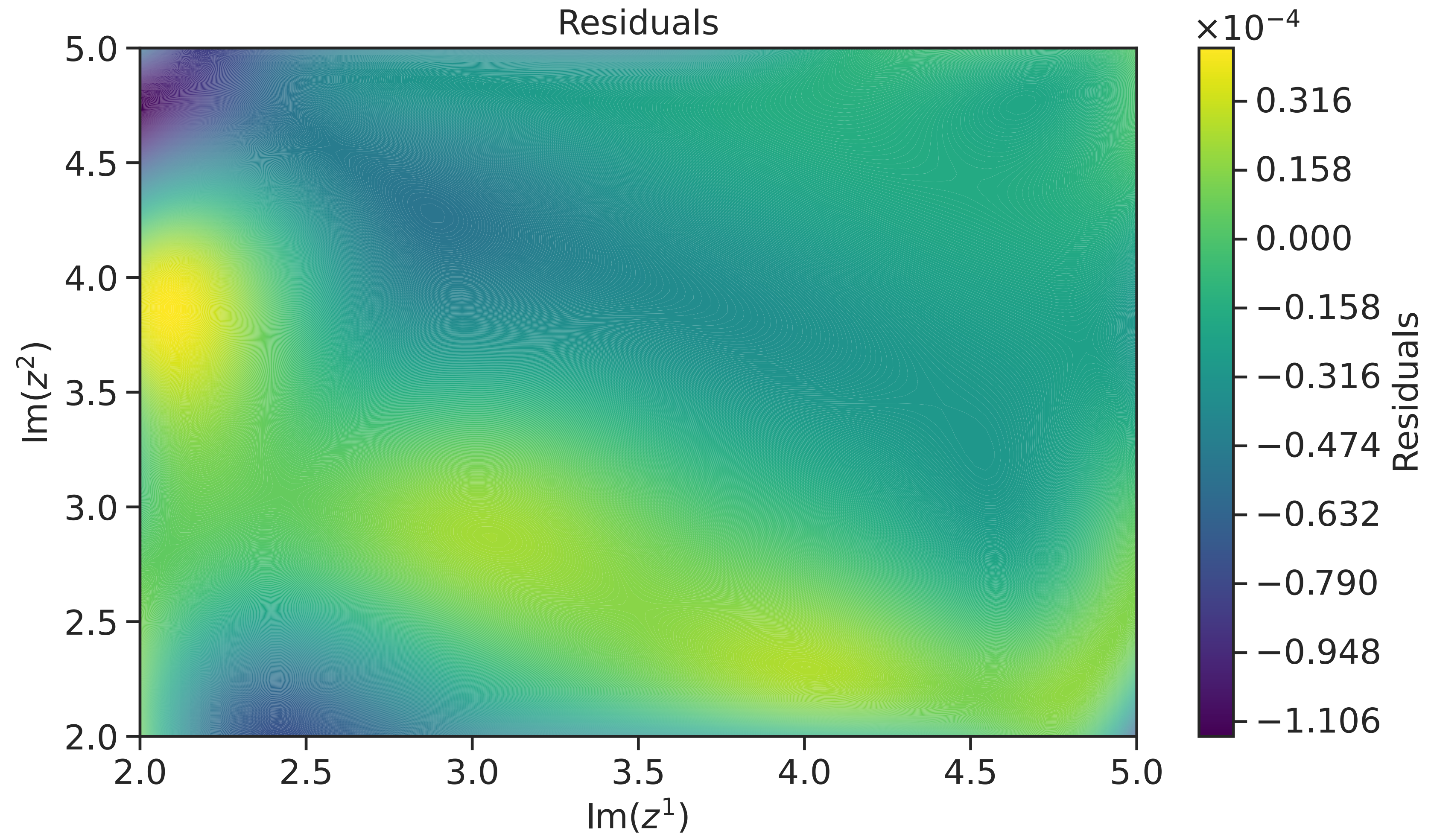


Continuous flux density expectation



Observed flux density





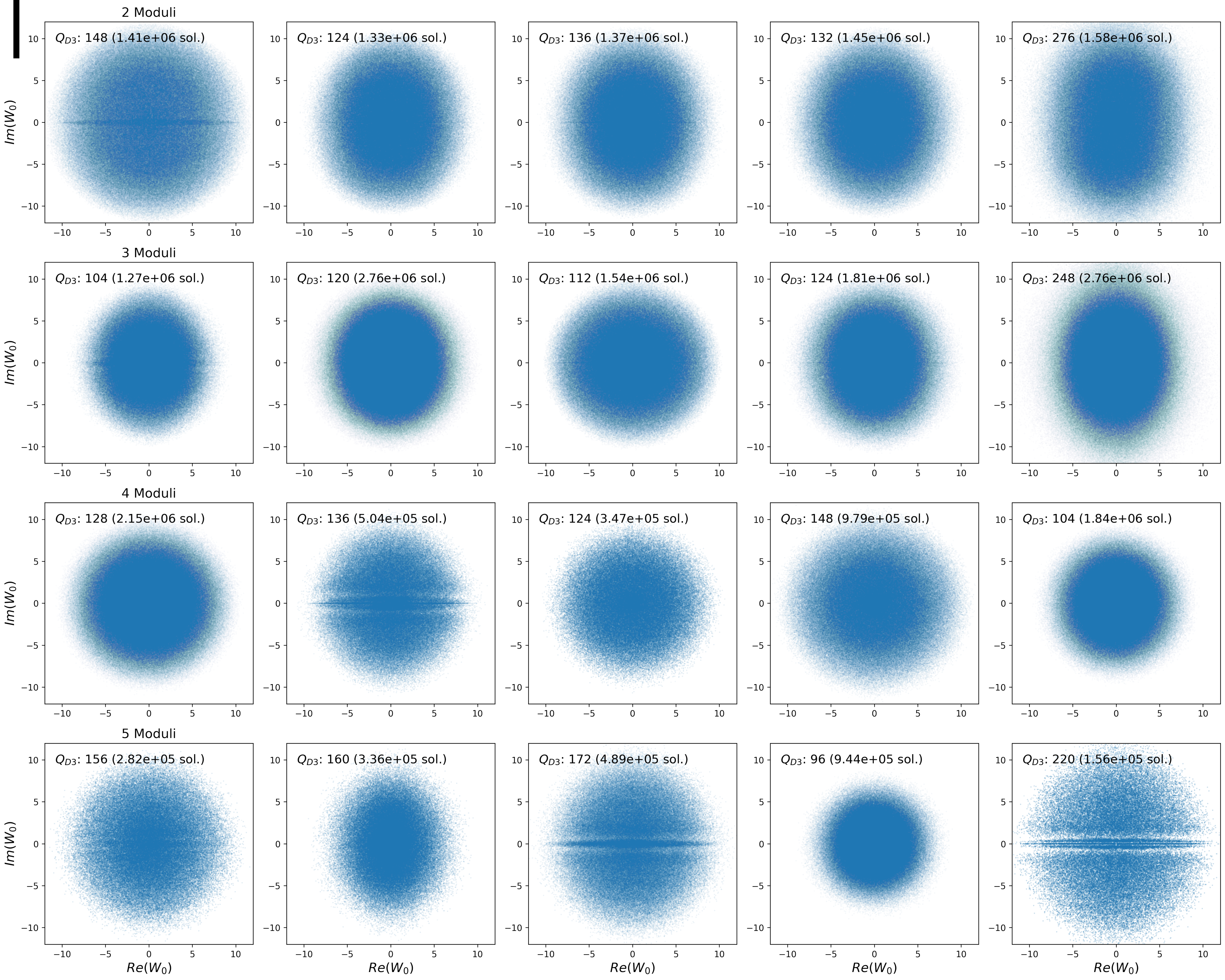
$$\mathcal{N}_{\text{stat}}(N_{\text{flux}} \leq N_{\text{max}}) = \frac{(2\pi N_{\text{max}})^6}{6!} \int_{\mathcal{M}_\tau \times \mathcal{M}_{\text{CS}}} \mathrm{d}^6 z \, \det(g) \, \rho(z)$$

Name	$\text{Im}(z^i)$	s	N_{max}	$\#h$	$\#f$	$\#(f,h)$	$\mathcal{N}_{\text{vac}}$	exhaustive
A	$[2,3]$	$[\frac{\sqrt{3}}{2}, 20]$	34	82,082	1,849,426	5,134,862	5,140,872	✓
B	$[2,5]$	$[\frac{\sqrt{3}}{2}, 10]$	10	1,900	6,340	12,160	12,196	✓
C	$[1,10]$	$[\frac{\sqrt{3}}{2}, 50]$	34	3,652,744	21,043,832	50,652,686	50,884,086	×
D	$[2,10]$	$[\frac{\sqrt{3}}{2}, 10]$	50	5,909,012	45,886,900	123,075,206	123,408,240	×

Distribution of $|W_0|$

- Coarse scan for various geometries showed universal behavior (2307.15747 with J. Ebelt).
- Structures around $\mathrm{Im}(W_0) = 0$ unclear.

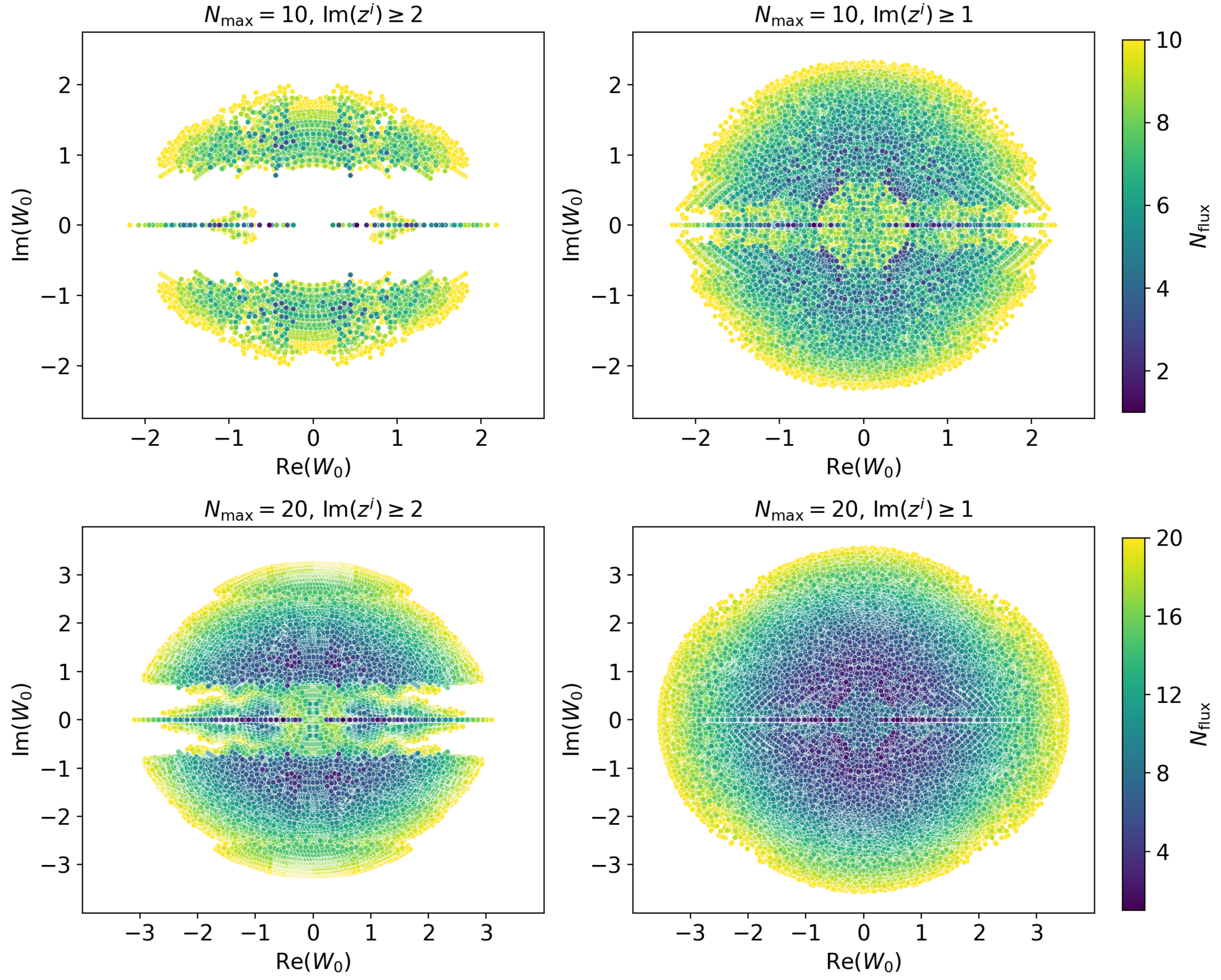
$$W_0 = \sqrt{2/\pi}\ e^{K/2}\ W$$

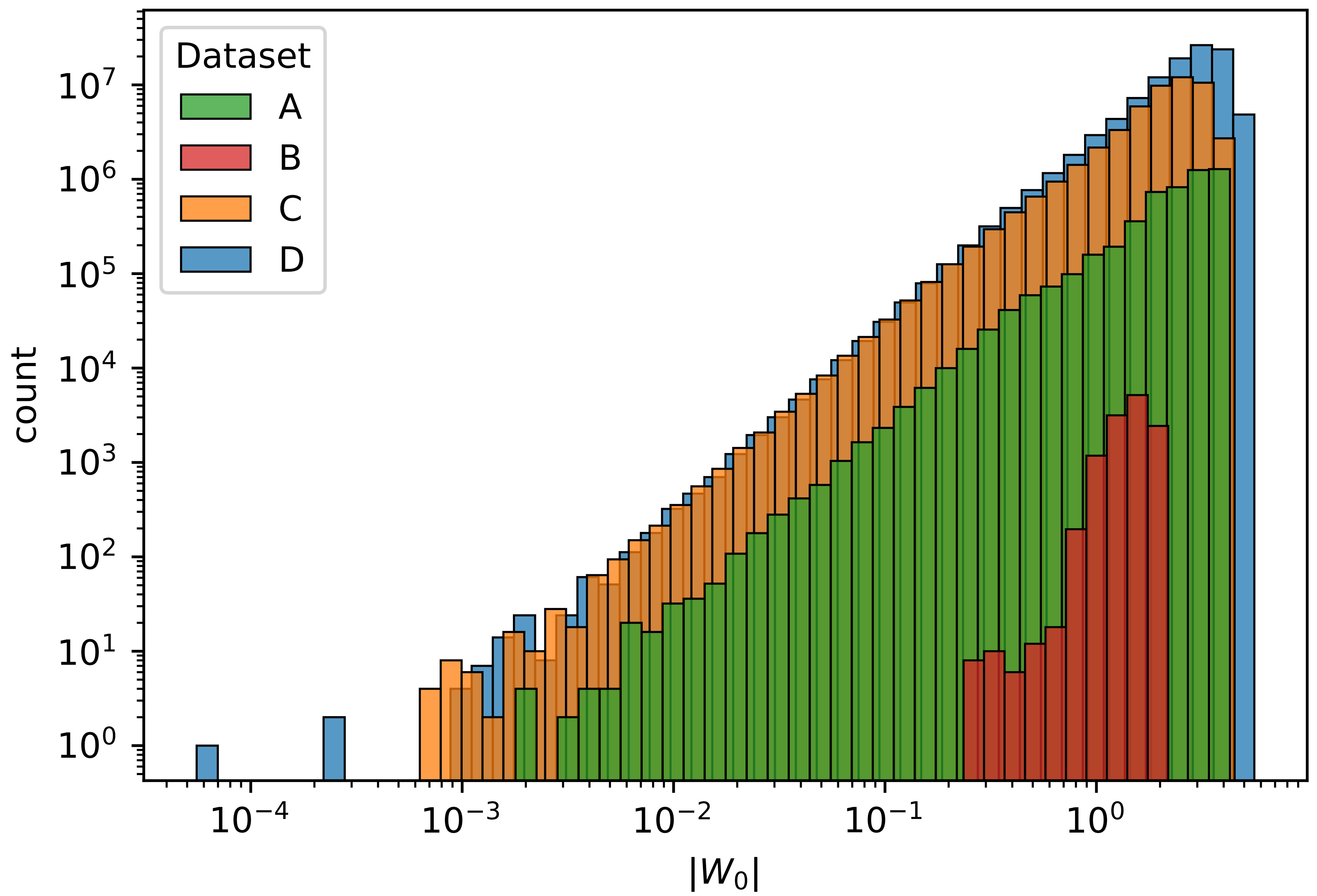


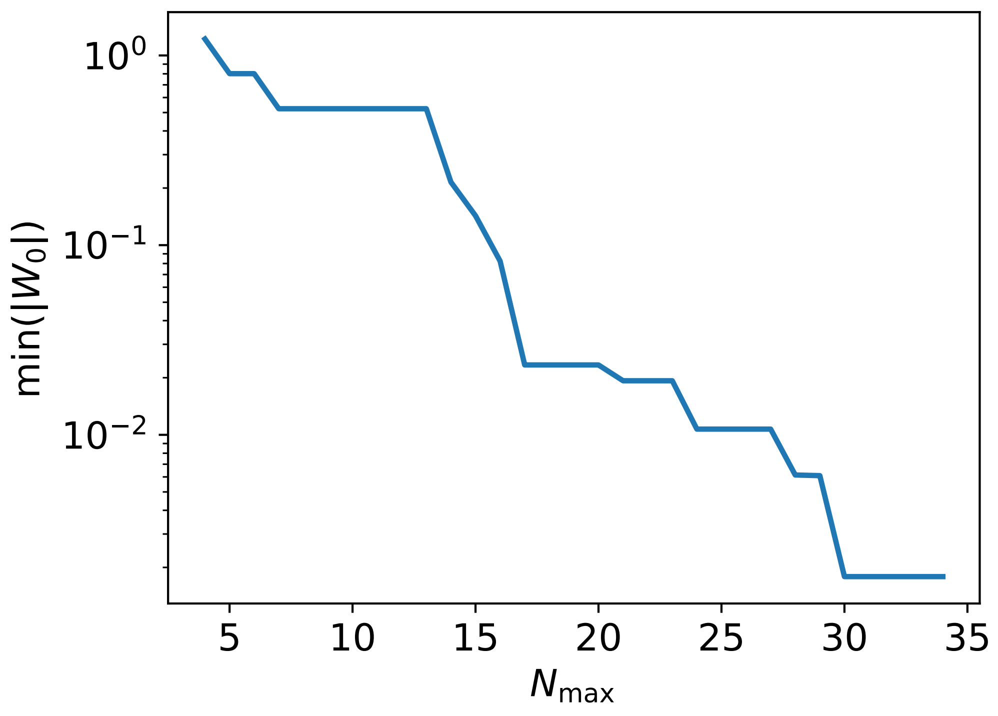
2307.15747

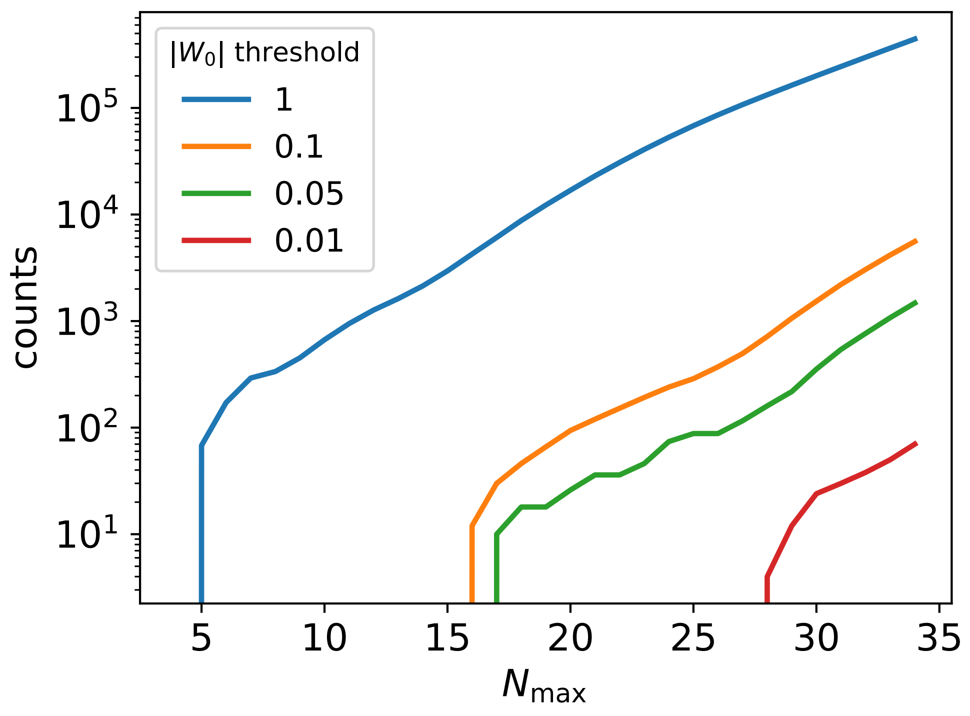
Distribution of $|W_0|$

Structures depend on sample construction (region & tadpole)

- 

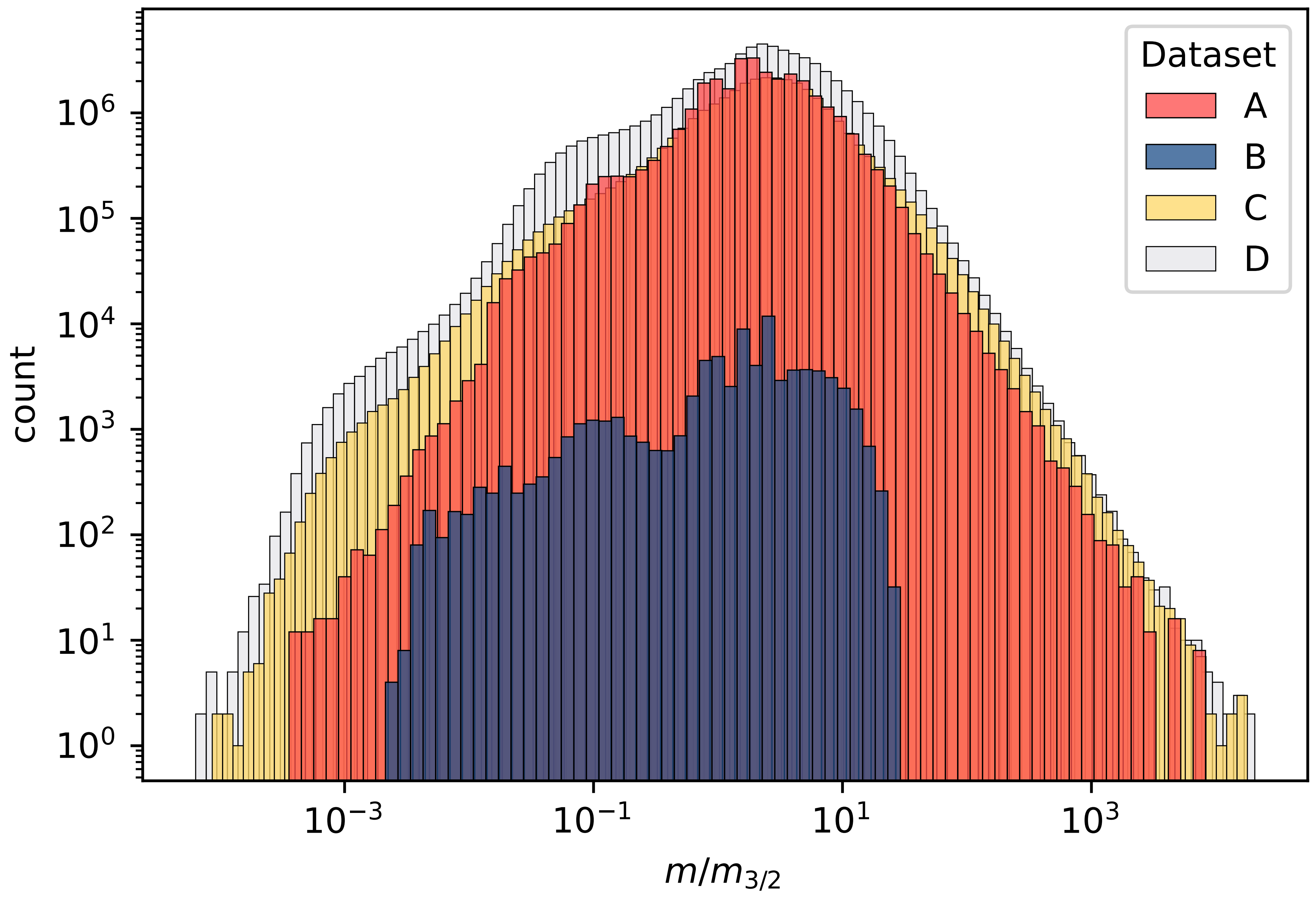


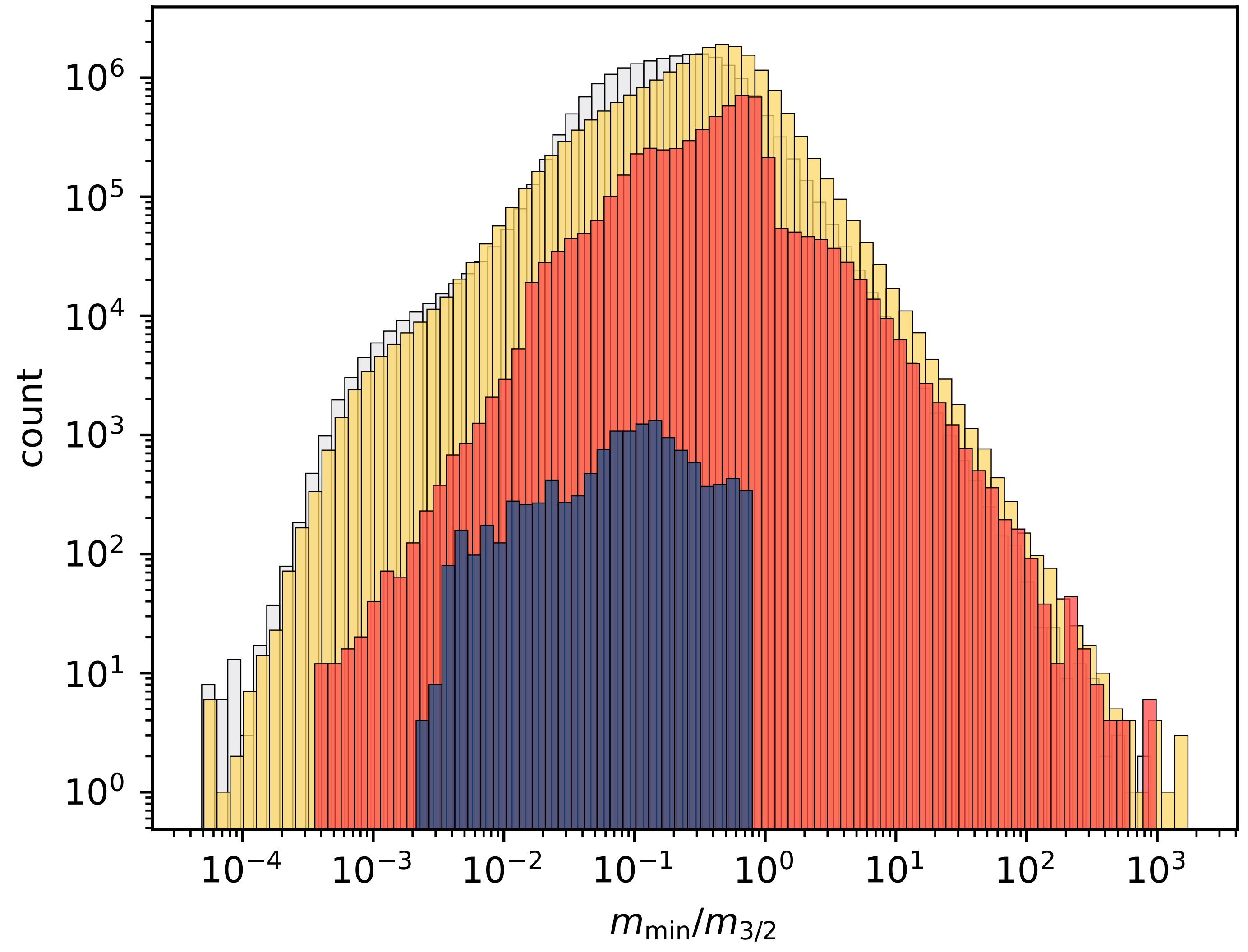


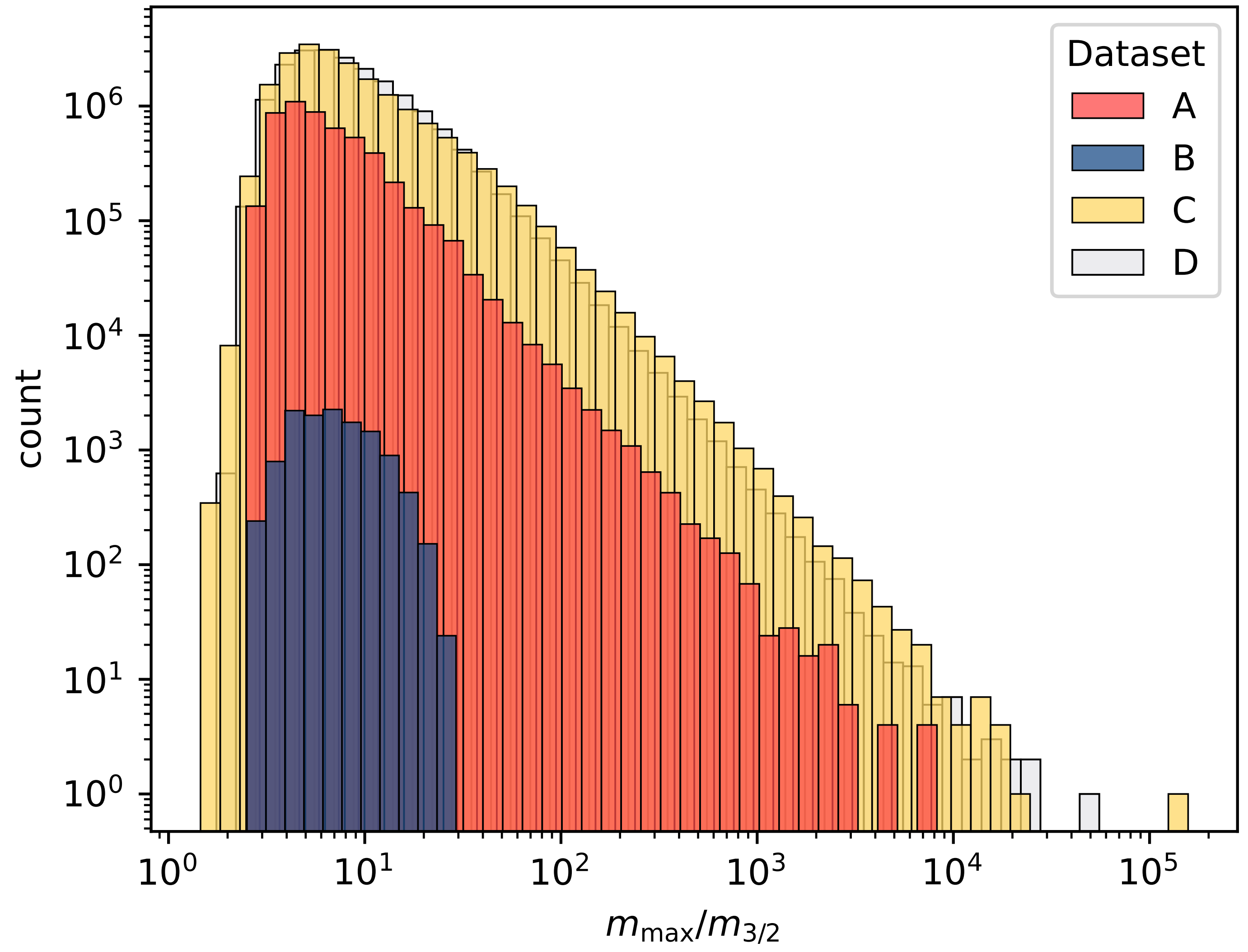


Distribution of moduli masses

Hierarchies are present







Explicit solution with: $|W_0| \sim 10^{-5}$

$$f = (4,12,2,-1,0,-1),\ h = (36,-1,0,0,1,-1)$$

$$z^1 = 0.5 + 2.3682\,\mathrm{i} z^2 = 0.5 + 2.5118\,\mathrm{i} \tau = 0.5 + 1.4812\,\mathrm{i}$$

$$m_A = (9.1505, 9.1513, 97.7826, 97.7853, 138.5255, 138.5287)$$

"Give me string models that realise $|W_0| = 100$."

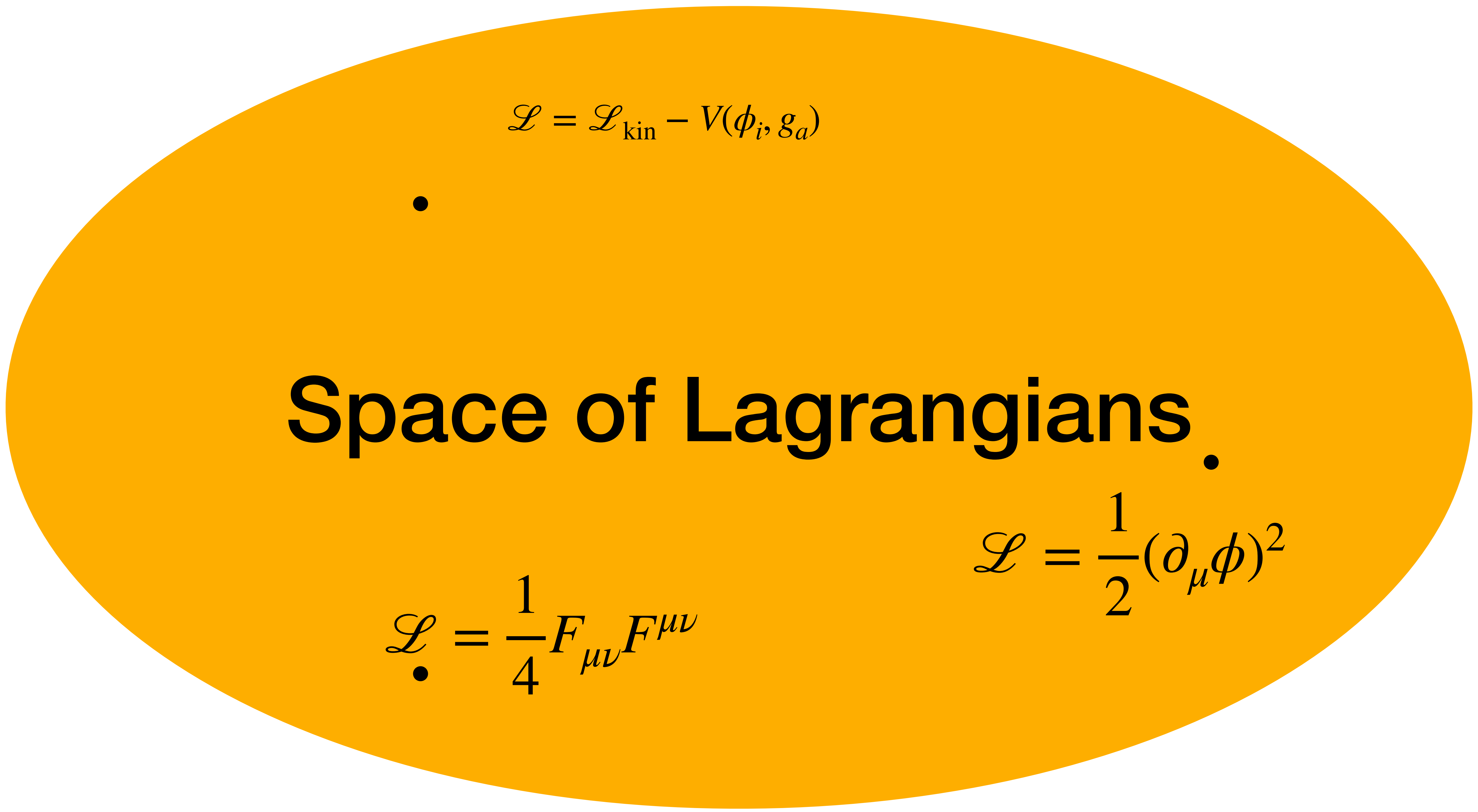
"What is the conditional density of flux vectors $P(\mathbf{x} \,|\, W_0)$?"

"What is the number of flux vacua with $|W_0| = 100$ and $N_{\text{flux}} < 10$?"

Elephant in the room: Scaling AI discoveries (to other string and BSM questions).

… wait for Part 3

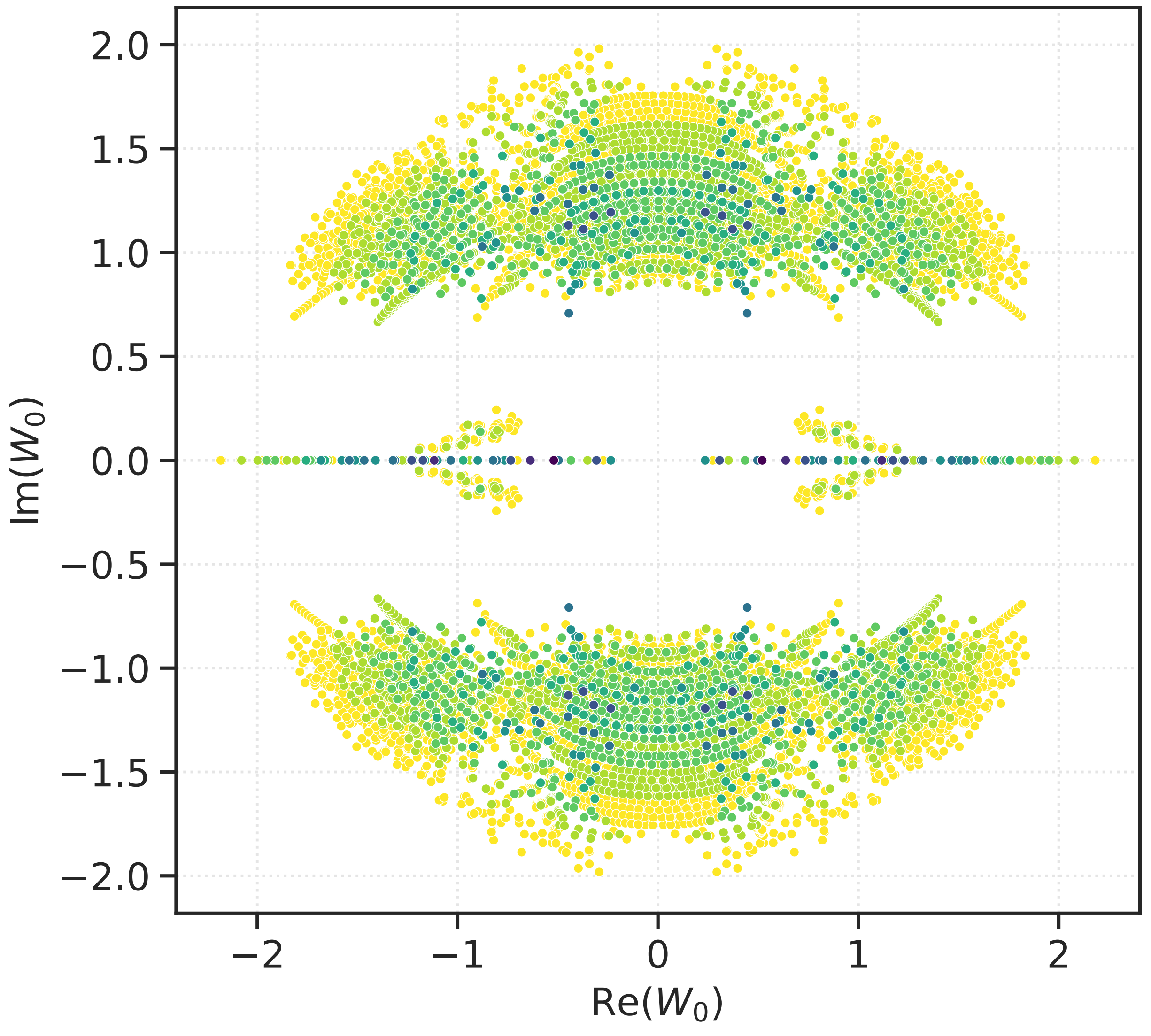
Conclusions Part 1

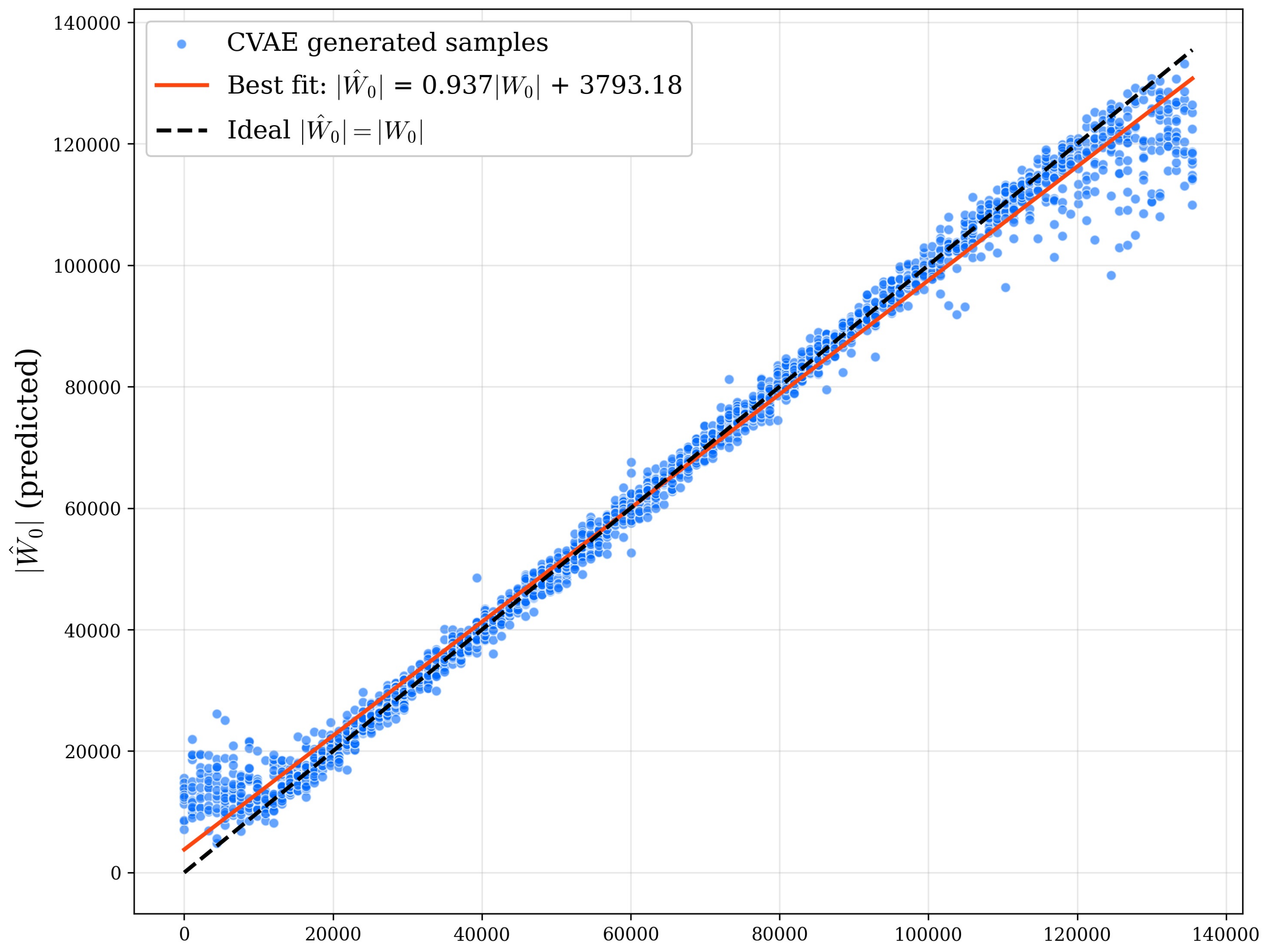


- Using customized ML models we can gain new insights in BSM physics:

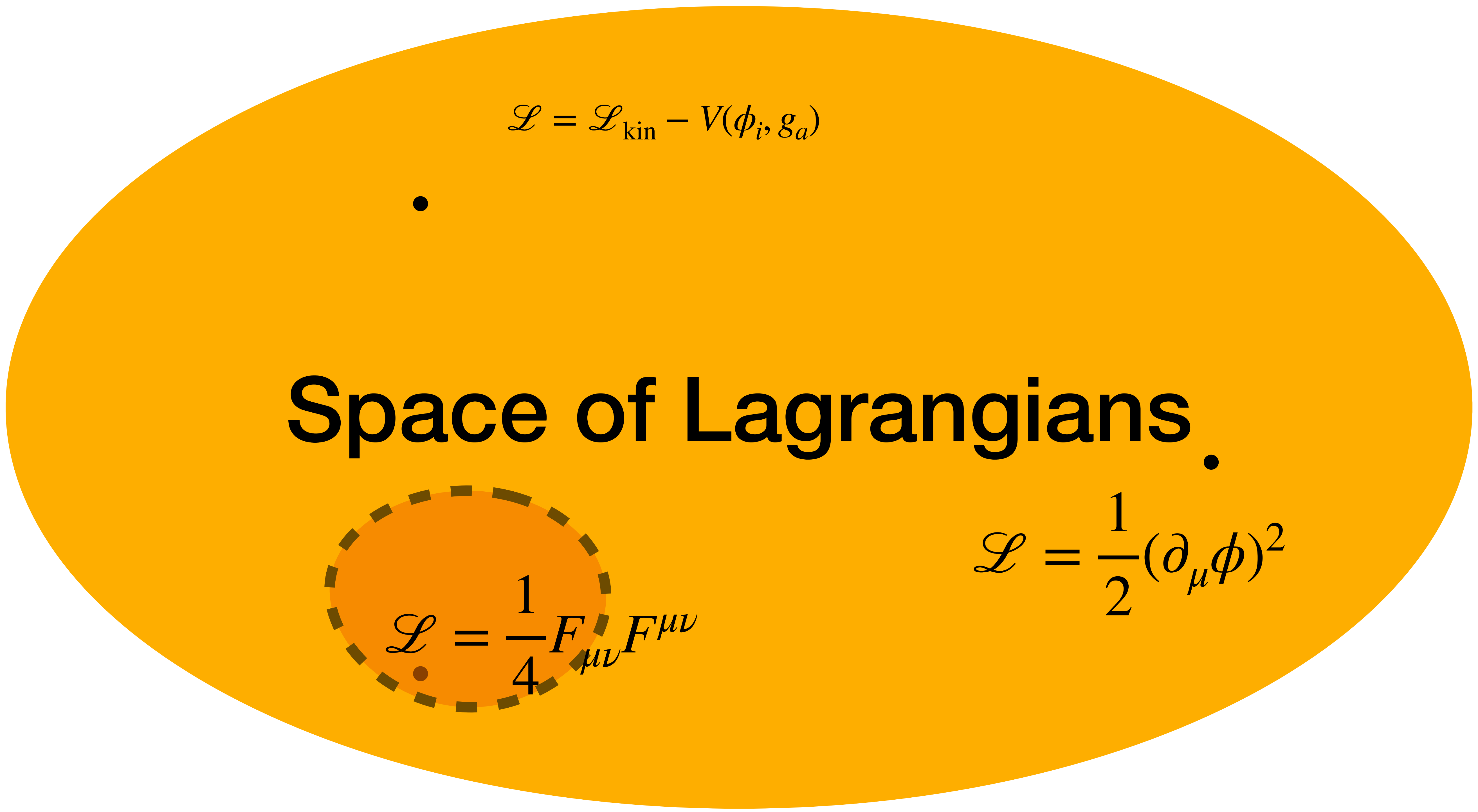
- Statistical models of string theory EFTs, e.g. $P(\mathbf{x} \,||\, W_0 |)$

- Deep observations of regions in the string landscape are possible.





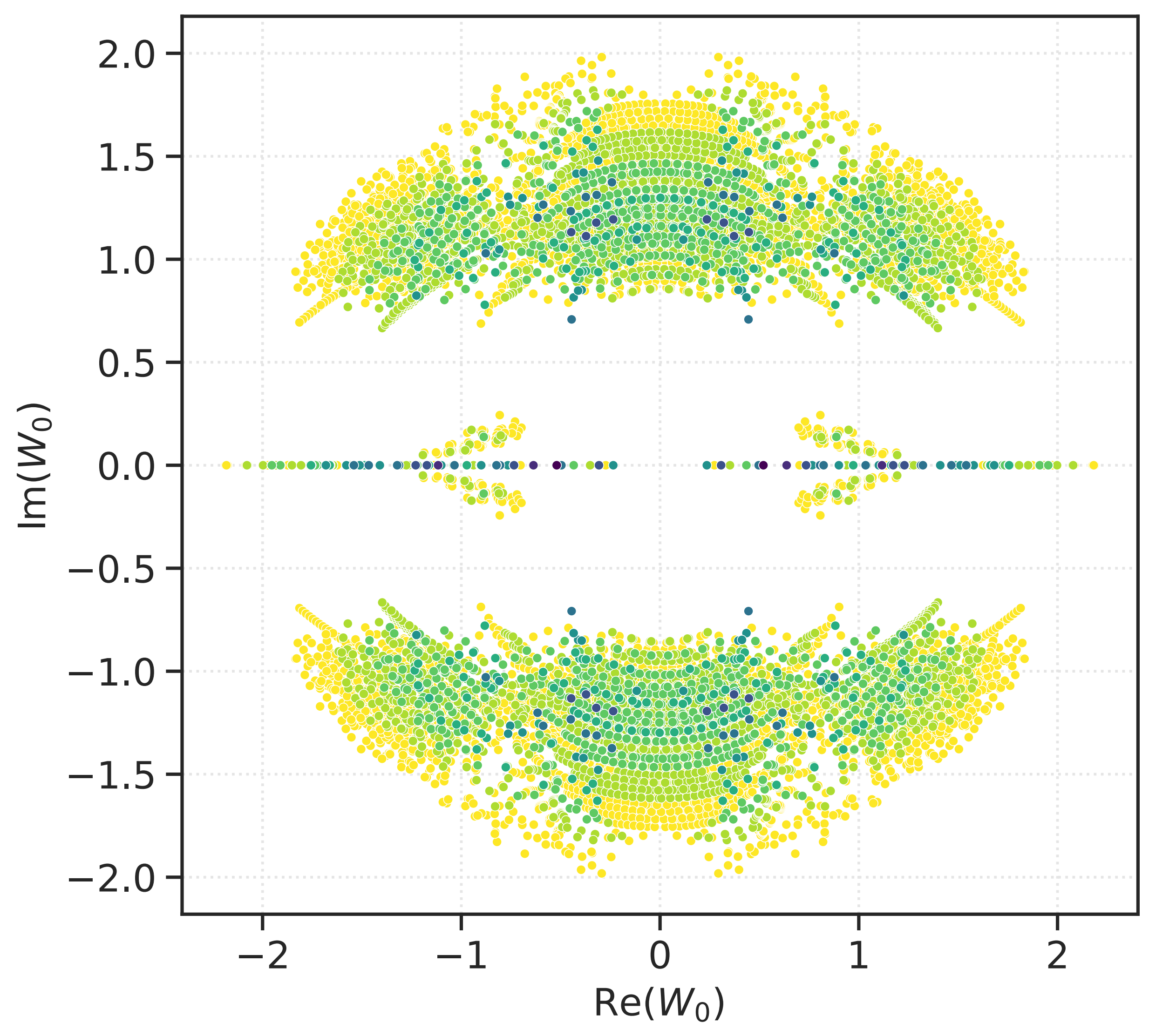
Conclusions Part 1

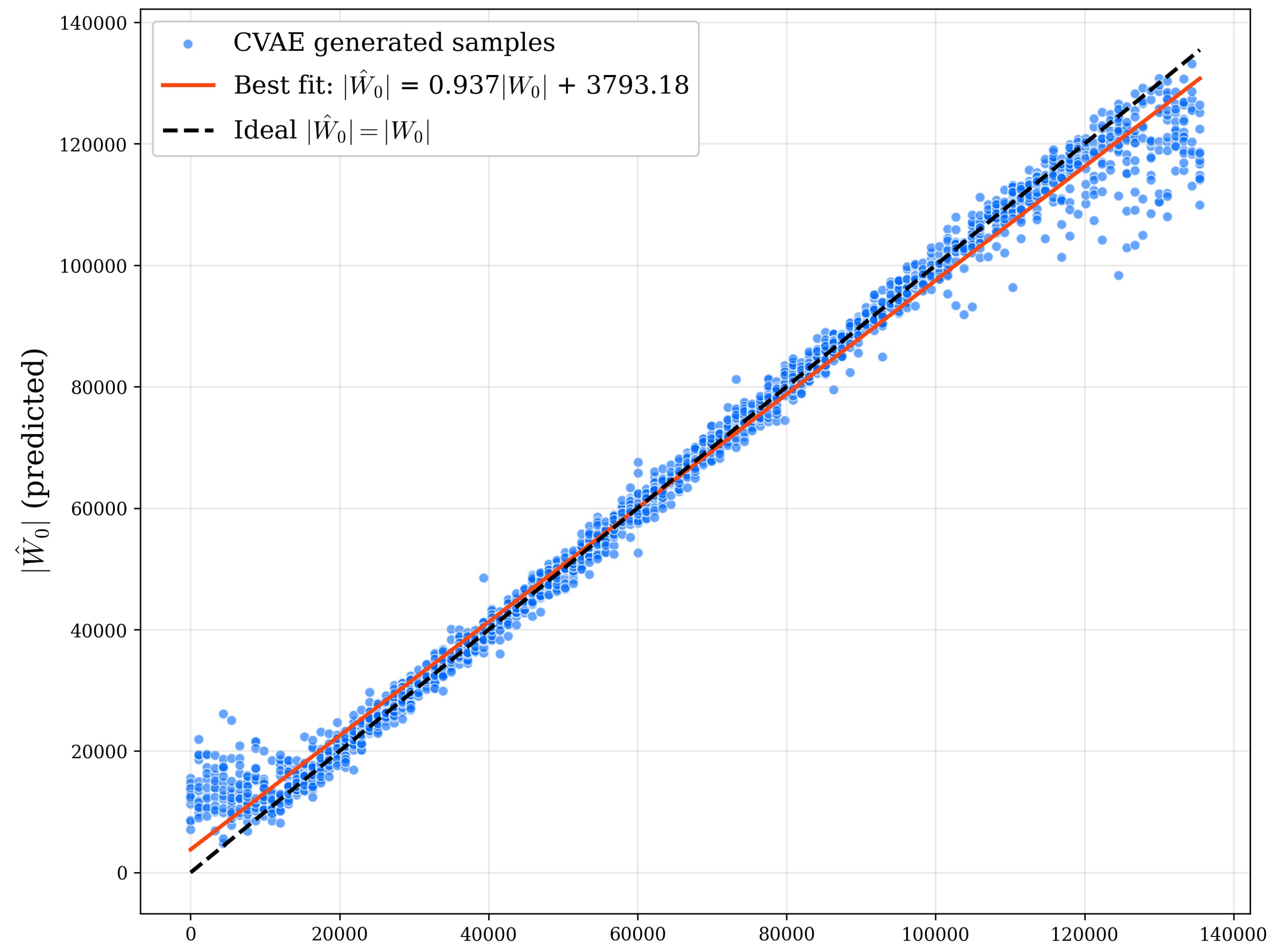


- Using customized ML models we can gain new insights in BSM physics:

- Statistical models of string theory EFTs, e.g. $P(\mathbf{x} \,||\, W_0 |)$

- Deep observations of regions in the string landscape are possible.





NN dynamics $\leftrightarrow$ Field Theory

based on 2202.11104 (MLST), 2305.00995 (MLST), and 2410.07451 (MLST), in collaboration with:

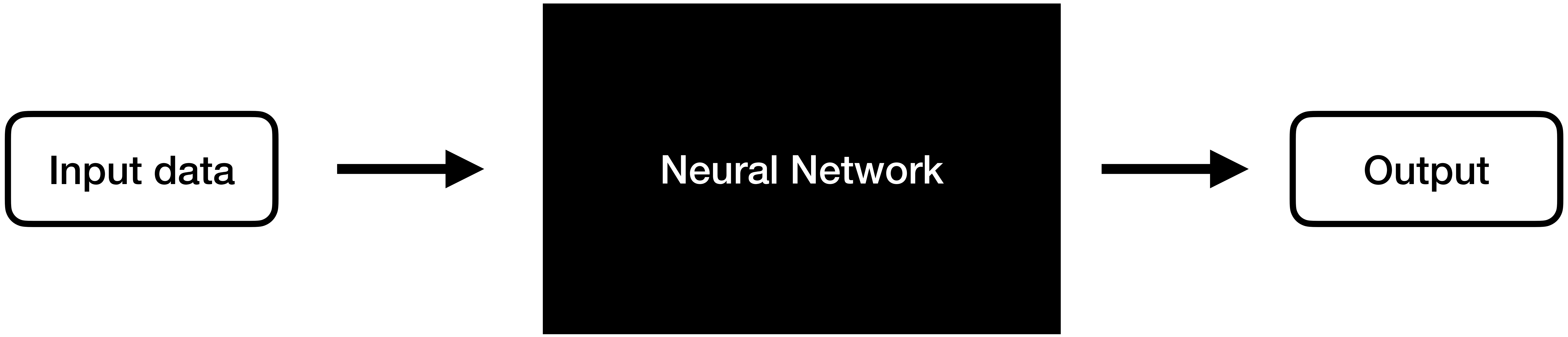
Michael Spannowsky

Sam Tovey

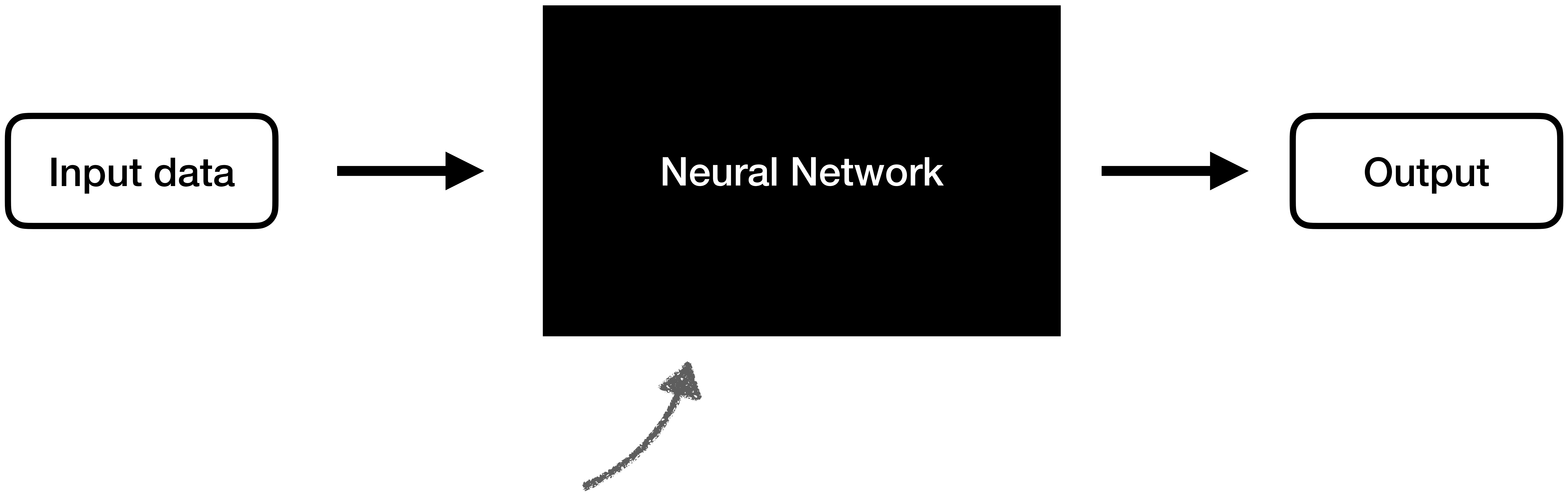
Konstantin Nikolaou

Christian Holm

Are neural networks black boxes?

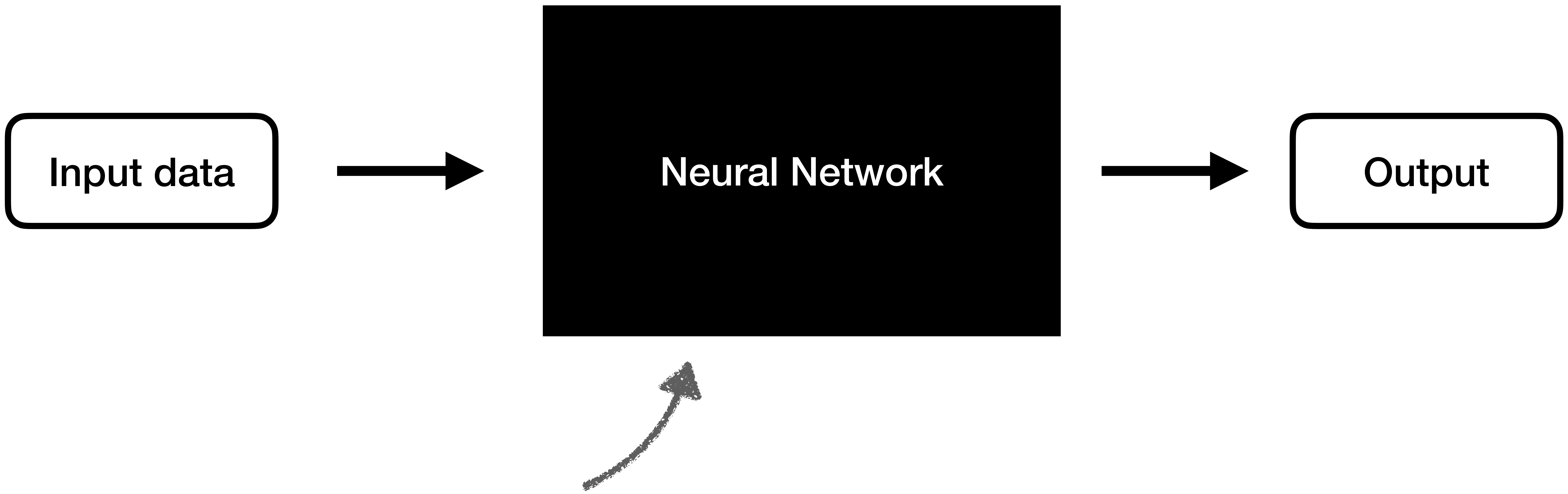


Are neural networks black boxes?



Analytic function, but many parameters so it’s not a simple function.

Are neural networks black boxes?



Analytic function, but many parameters so it’s not a simple function.

Do we know what is going on inside them?

Some hints: scaling laws

e.g. performance improves with more parameters

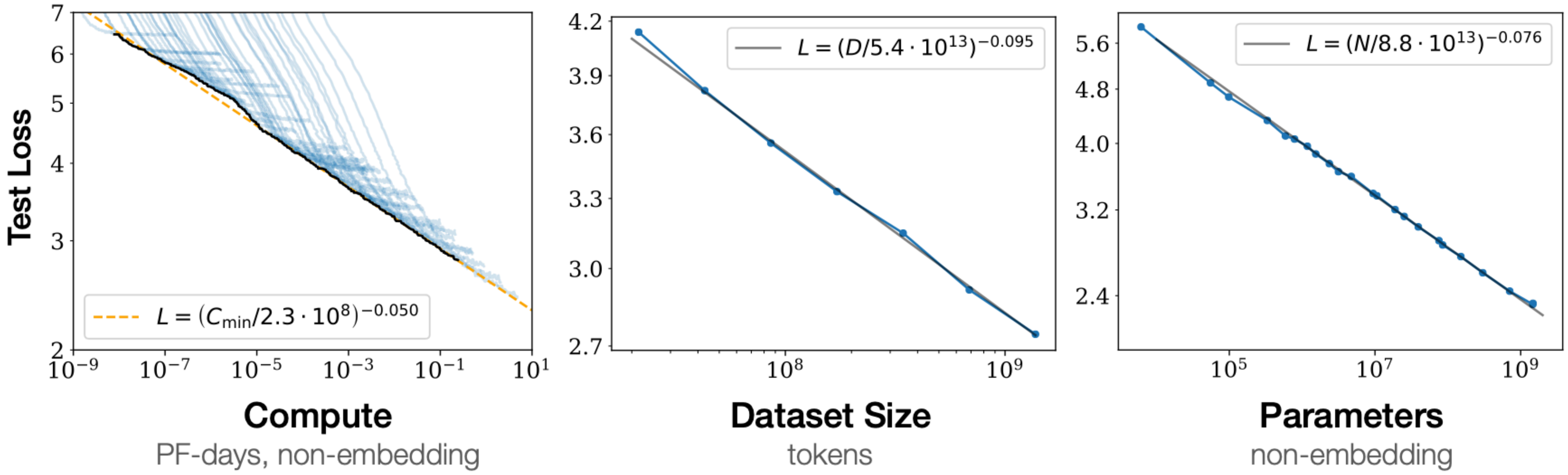


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

e.g. Kaplan et al 2020 Scaling laws for neural language models

We were able to precisely model the dependence of the loss on N and D, and alternatively on N and S, when these parameters are varied simultaneously. We used these relations to derive the compute scaling, magnitude of overfitting, early stopping step, and data requirements when training large language models. So our scaling relations go beyond mere observation to provide a predictive framework. One might interpret these relations as analogues of the ideal gas law, which relates the macroscopic properties of a gas in a universal way, independent of most of the details of its microscopic consituents.

It is natural to conjecture that the scaling relations will apply to other generative modeling tasks with a maximum likelihood loss, and perhaps in other settings as well. To this purpose, it will be interesting to test these relations on other domains, such as images, audio, and video models, and perhaps also for random network distillation. At this point we do not know which of our results depend on the structure of natural language data, and which are universal. It would also be exciting to find a theoretical framework from which the scaling relations can be derived: a 'statistical mechanics' underlying the 'thermodynamics' we have observed. Such a theory might make it possible to derive other more precise predictions, and provide a systematic understanding of the limitations of the scaling laws.

Kaplan et al 2020 Scaling laws for neural language models

We were able to precisely model the dependence of the loss on N and D, and alternatively on N and S, when these parameters are varied simultaneously. We used these relations to derive the compute scaling, magnitude of overfitting, early stopping step, and data requirements when training large language models. So our scaling relations go beyond mere observation to provide a predictive framework. One might interpret these relations as analogues of the ideal gas law, which relates the macroscopic properties of a gas in a universal way, independent of most of the details of its microscopic consituents.

It is natural to conjecture that the scaling relations will apply to other generative modeling tasks with a maximum likelihood loss, and perhaps in other settings as well. To this purpose, it will be interesting to test these relations on other domains, such as images, audio, and video models, and perhaps also for random network distillation. At this point we do not know which of our results depend on the structure of natural language data, and which are universal. It would also be exciting to find a theoretical framework from which the scaling relations can be derived: a 'statistical mechanics' underlying the 'thermodynamics' we have observed. Such a theory might make it possible to derive other more precise predictions, and provide a systematic understanding of the limitations of the scaling laws.

Do we know what is going inside NNs?

For us becomes: Theoretical framework to quantify dynamical behaviour of NNs?

Kaplan et al 2020 Scaling laws for neural language models

Physics to understand NN dynamics

Problems and our approach

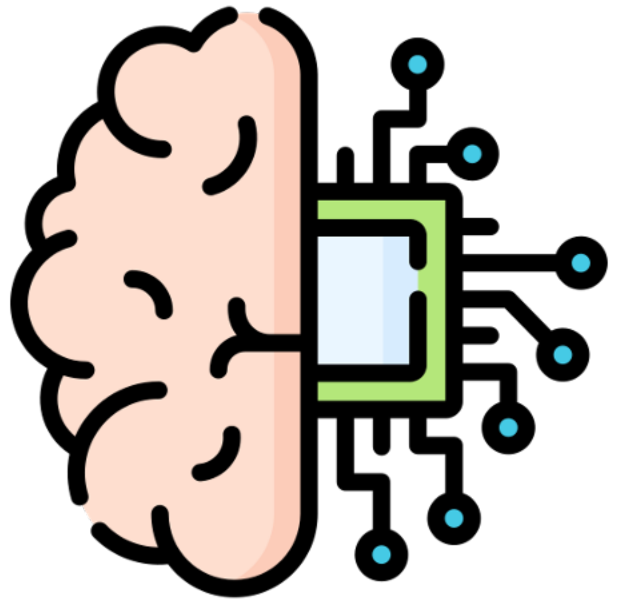
Parameters	175 billion
Training Time	Several months
Training Cost	~ $4.6 million

OpenAI

- We cannot afford hyperparameter scans for such large networks. *How to successfully predict training performance?*

Neurons	86 billion
Object recognition time[2]	150 ms
Energy cost[1]	< 20 W

[1] (Sterling & Laughlin, 2015), [2] (Thorpe et al., 1996)



- Our NN networks are not energy efficient. *How to improve efficiency of NNs to make them useful with less computational resources?*

cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758

Physics to understand NN dynamics

Problems and our approach

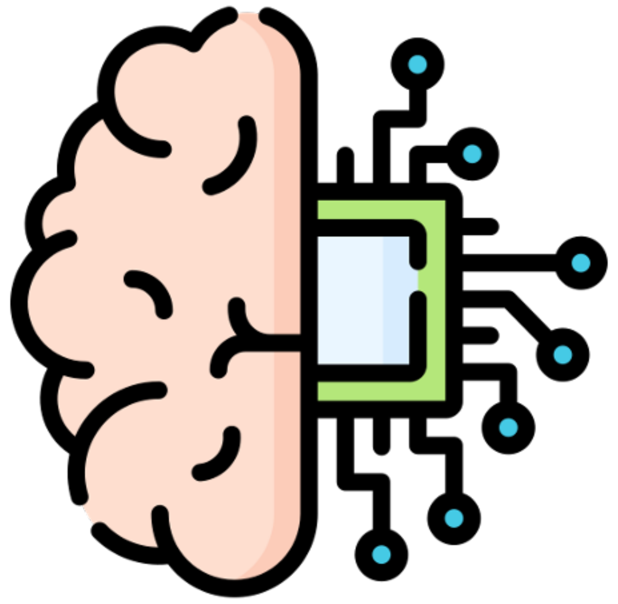
Parameters	175 billion
Training Time	Several months
Training Cost	~ $4.6 million

OpenAI

- We cannot afford hyperparameter scans for such large networks. *How to successfully predict training performance?*

Neurons	86 billion
Object recognition time[2]	150 ms
Energy cost[1]	< 20 W

[1] (Sterling & Laughlin, 2015), [2] (Thorpe et al., 1996)



- Our NN networks are not energy efficient. *How to improve efficiency of NNs to make them useful with less computational resources?*

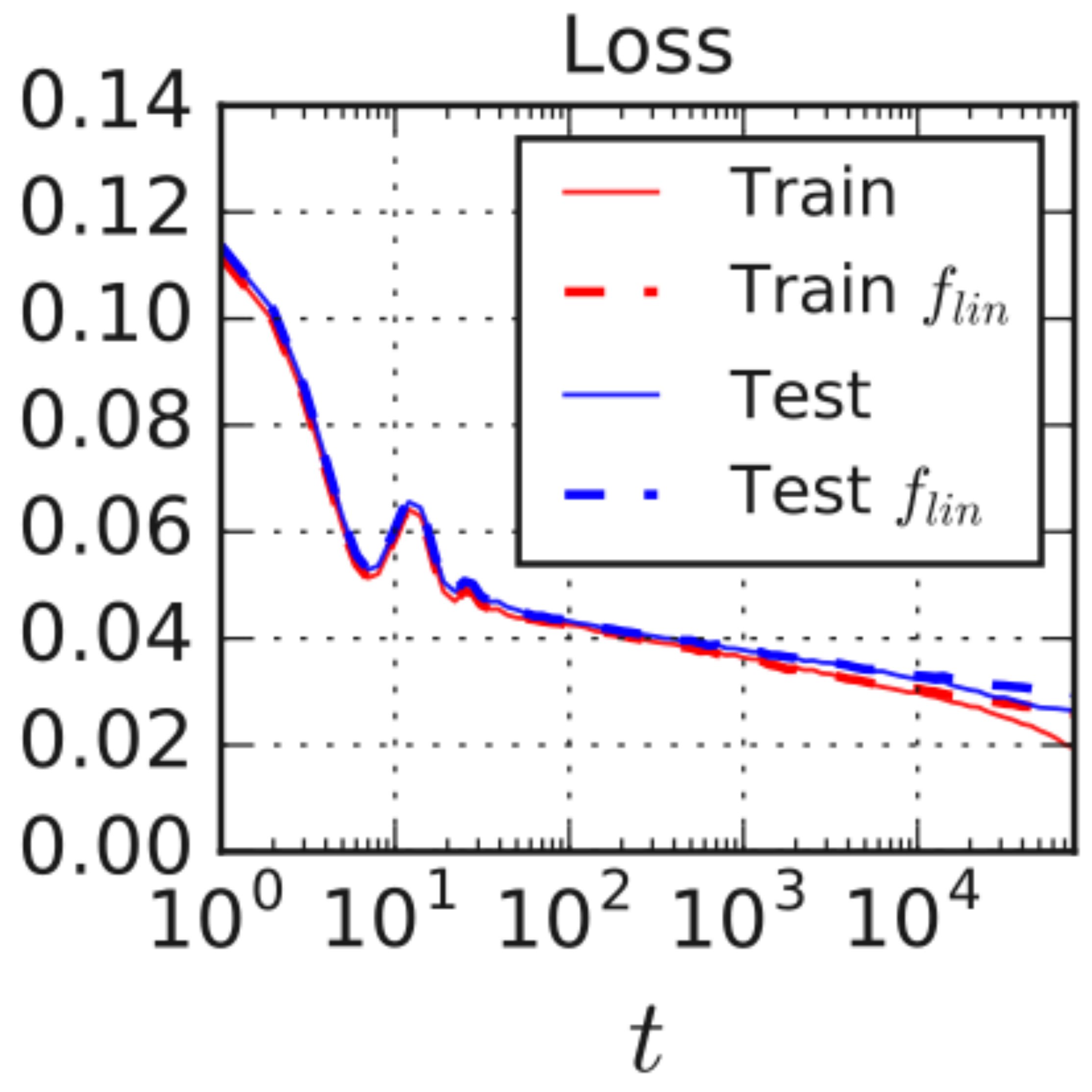
cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758

Describe neural networks & dynamics via dynamics of collective variables.
Aim: control and improve learning of NNs.

How do we link dynamics of NNs and collective variables?

Understand NN dynamics via empirical NTK

Simplification of dynamics in large width limit



- The dynamics of a neural network $f(x, \theta)$ simplify in the infinite width limit.
- The NN equations in continuous time limit:

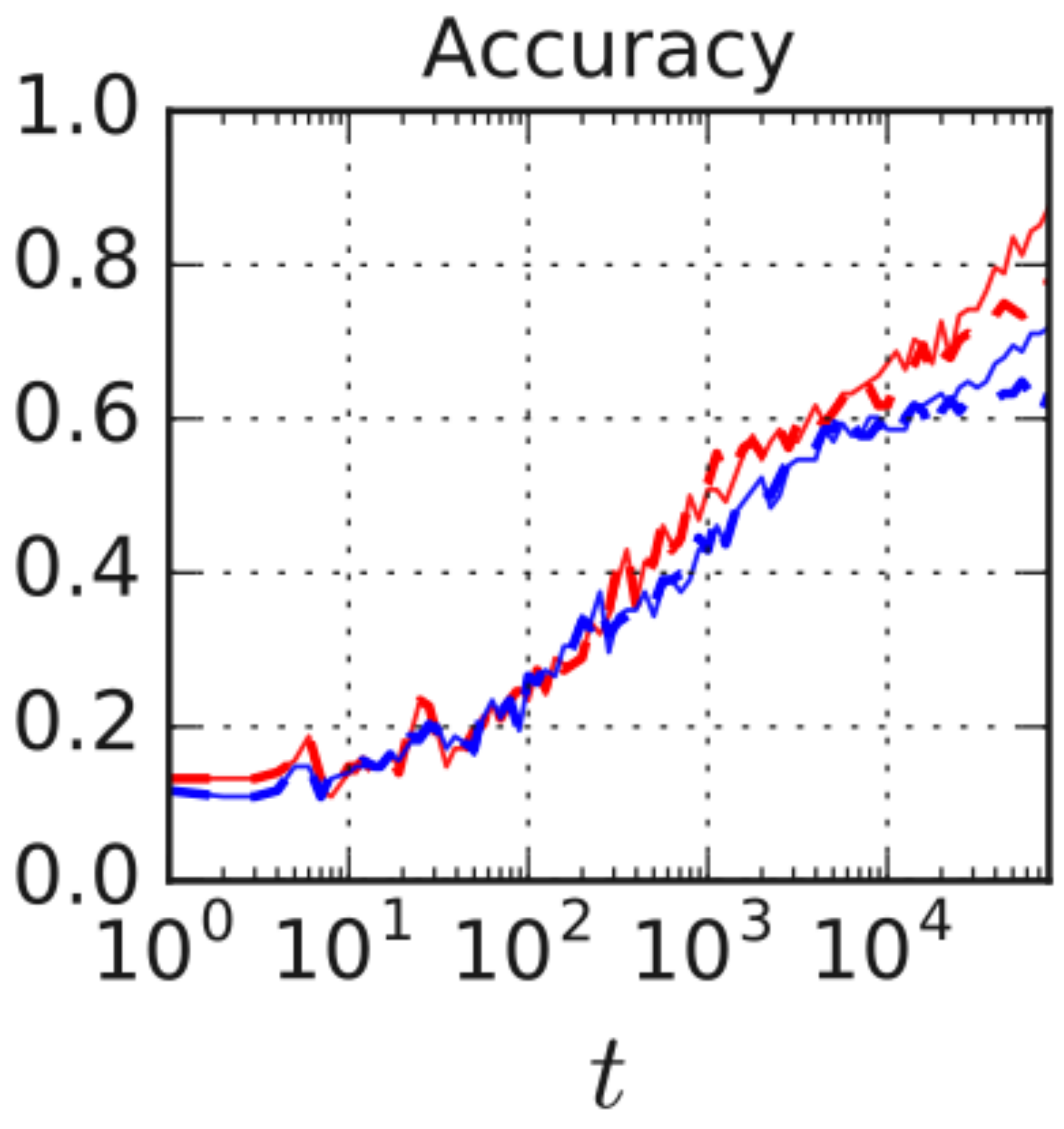
$$\dot{\theta} = -\eta \nabla_\theta \mathscr{L} = -\eta \nabla_\theta f(y) \nabla_{f(y)} \mathscr{L}$$

$$\dot{f}(x) = \nabla_\theta f(x)\ \dot{\theta} = -\eta \nabla_\theta f(x) \nabla_\theta f(y) \nabla_{f(y)} \mathscr{L} = -\eta \Theta(x, y) \nabla_{f(y)} \mathscr{L}$$

- NN update simplify in large width limit: Neural tangent kernel remains constant (empirical and analytical):

$$\Theta(t, x, y) = \Theta(t = 0, x, y)$$

- Complete as all learning components included: finite data, optimisers, and NN architecture
- Not sufficient (e.g. not capturing feature learning), in practice $\Theta(t, x, y) \approx \Theta(t = 0, x, y)$ at finite but large width. Which simple model describes the dynamics of NTK?



Wide resnet trained by SGD with momentum on CIFAR-10 (from 1902.06720)

Krippendorf, Spannowsky: 2202.11104
Tovey, Krippendorf, Nikolou, Holm: 2305.00995

Jacot, Gabrial, Hongler
Lee, Xiao, Schoenholz, Bahri, Novak, Sohl-Dickstein, Pennington
Novak, Xiao, Hron, Lee, Alemi, Sohl-Dickstein, Schoenholz

Scales in NN dynamics

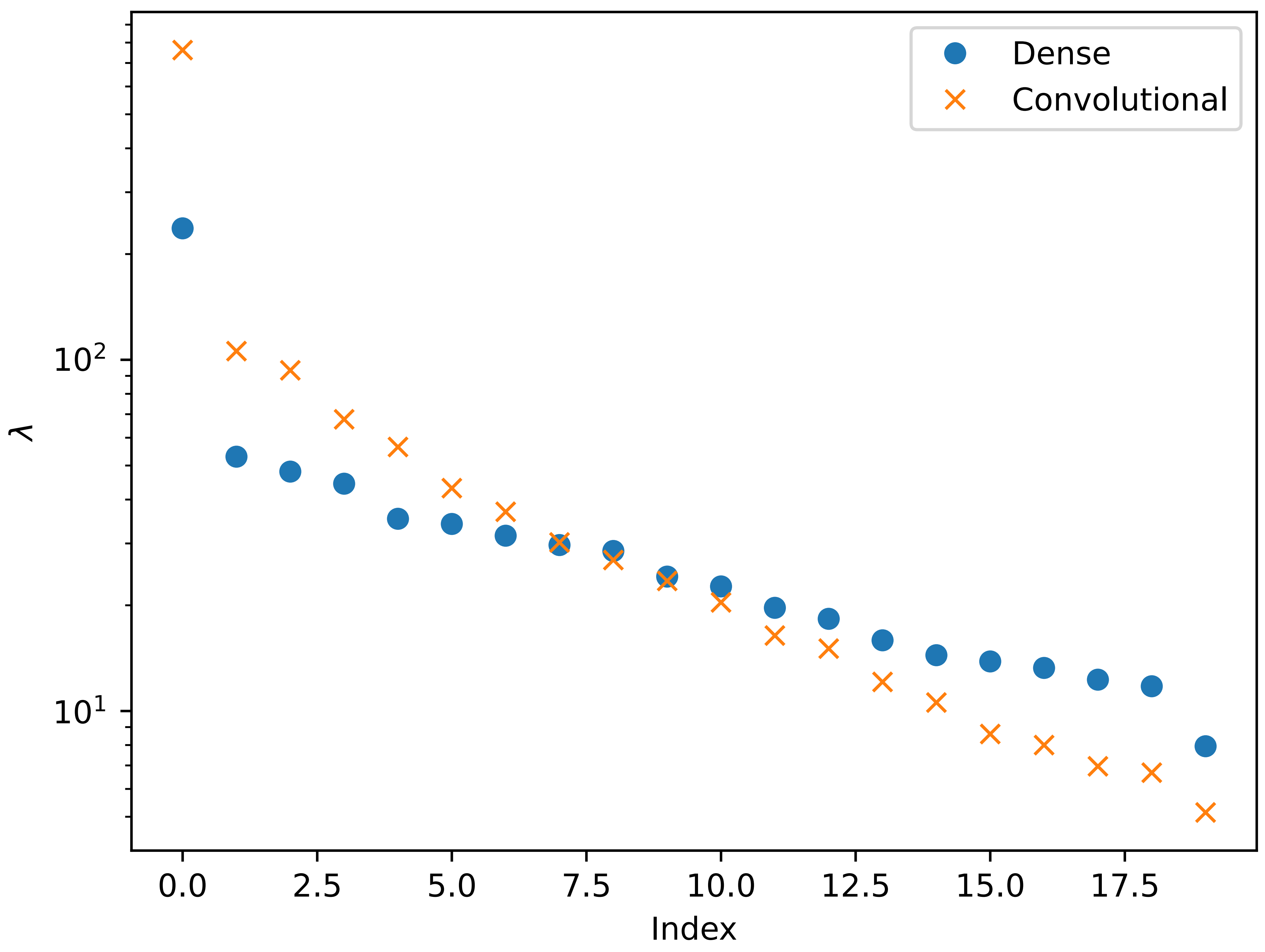
Hierarchical spectrum in NTK → EFT (coll. variable) approach promising

- Diagonalise NTK ($\Theta_{\rm NTK}$) NN-update equation:

$$\dot{\tilde{f}}(\mathscr{D}) = -\eta \ \mathrm{diag}(\lambda_1, \ldots, \lambda_{\rm N}) \ \mathscr{L}'(\mathscr{D})$$

- Largest changes in modes with largest eigenvalues.
- Hierarchical spectrum in NTK, consequences:
  - Effectively dynamics take place in lower-dimensional subspace.
  - There are few "collective" variables in NTK which determine the dynamics. Their time evolution is what we need to understand.
  - Limit: adding more data does not change dynamics if non-vanishing eigenvalues are not changed (naturally cut-offs do appear in analogy with effective field theories).

cf. Gur-Ari, Roberts, Dyer 2018



spectrum perspective: 2202.11104 (MLST)

EFT and data points

EFT and data points

- #data points (N) sets # modes as empirical NTK is N x N matrix:

$$\nabla_\theta f(x_i)\,\nabla_\theta f(x_j)$$

- Choice of data points sets which modes are chosen.

EFT and data points

- #data points (N) sets # modes as empirical NTK is N x N matrix:

$$\nabla_\theta f(x_i)\,\nabla_\theta f(x_j)$$

- Choice of data points sets which modes are chosen.

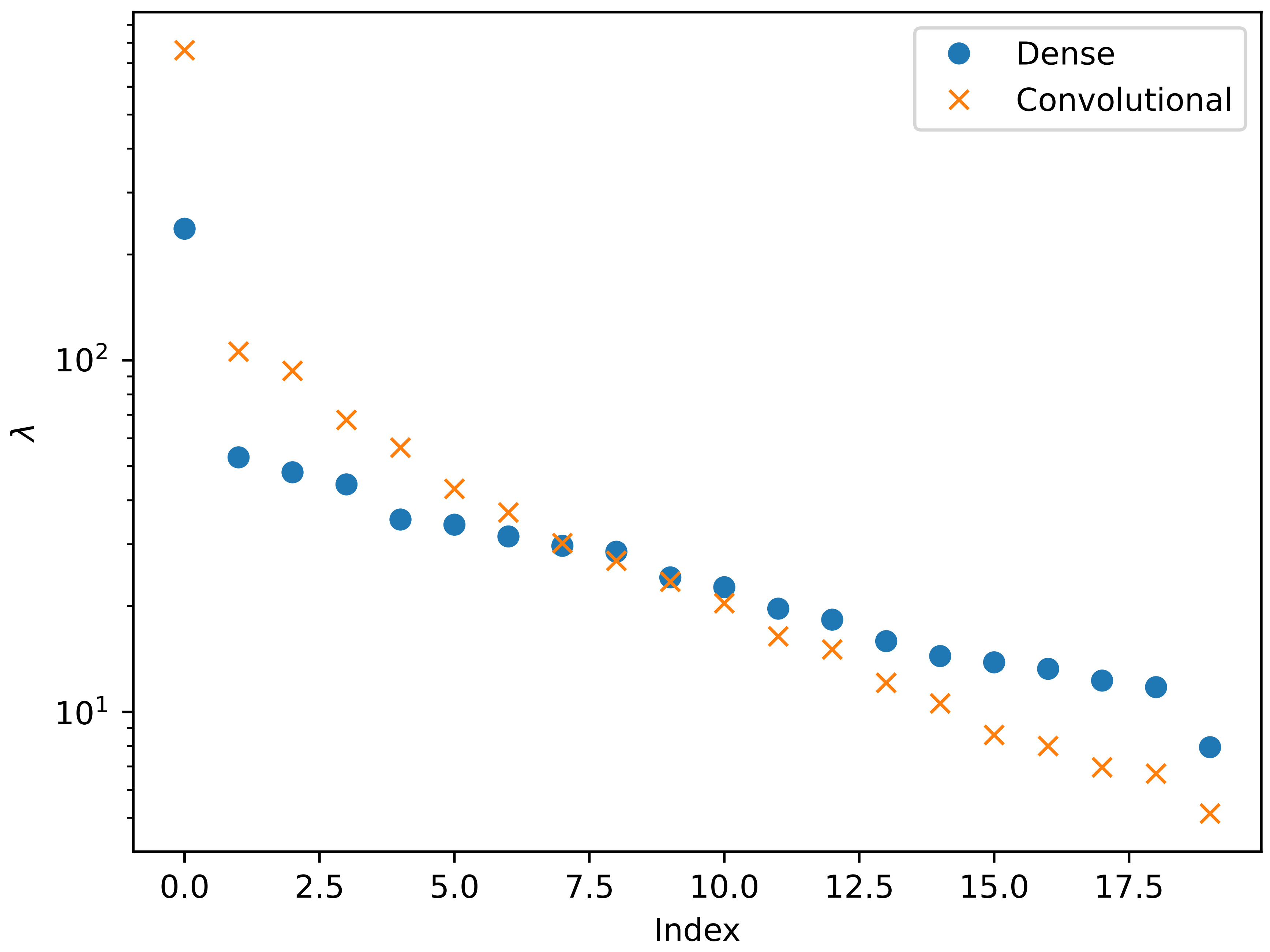
    - **Can we make training more data efficient by selecting good data points?**

$$\dot{f}(x) = -\eta\,\boxed{\nabla_\theta f(x)\,\nabla_\theta f(y)}\,\nabla_{f(y)}\mathscr{L}$$

important for update!

Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

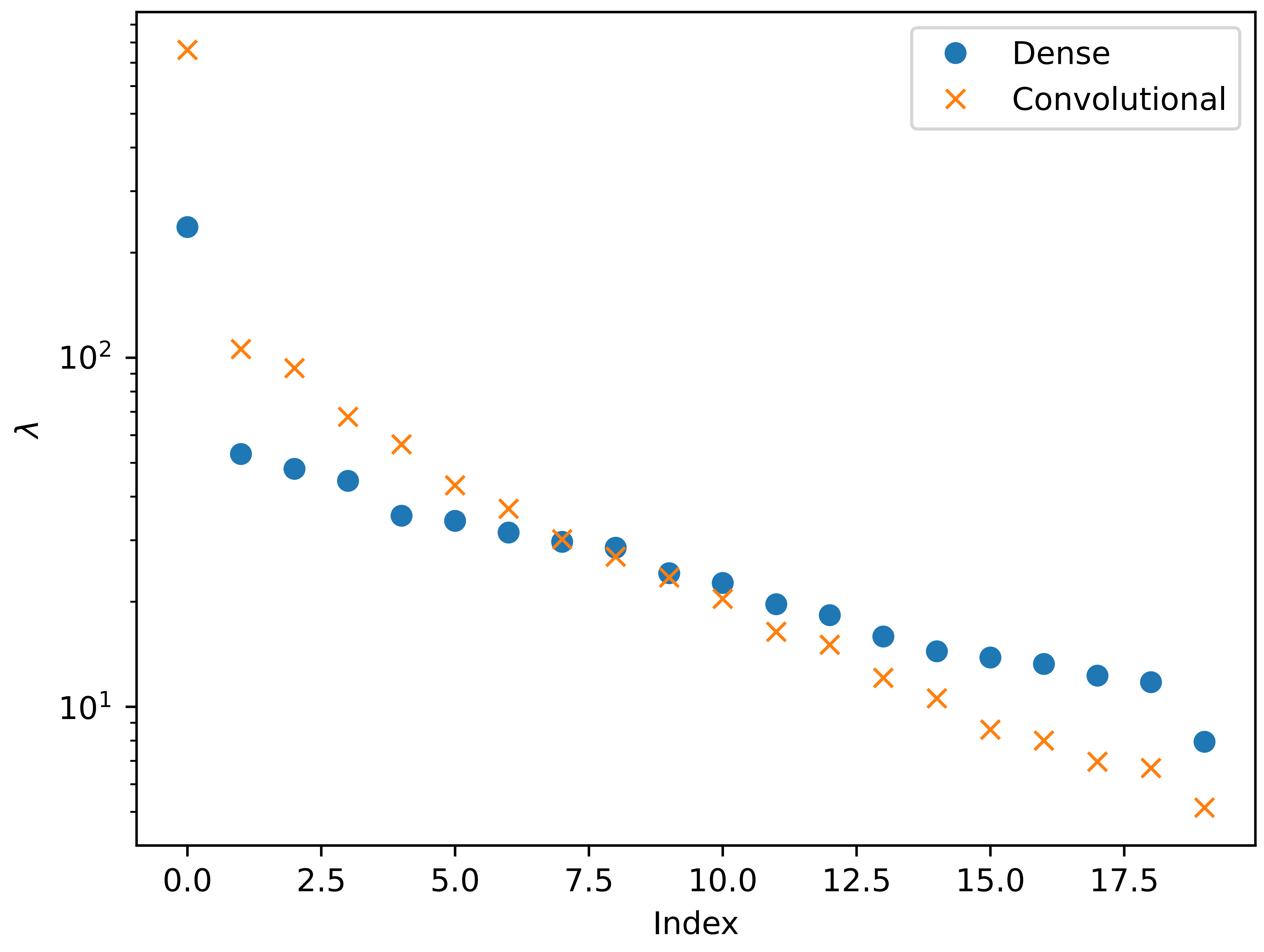


Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

- We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

$$\mathrm{Tr}(\Theta_{\mathrm{NTK}}) = \sum_i \lambda_i \approx \lambda_{\max}$$



Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

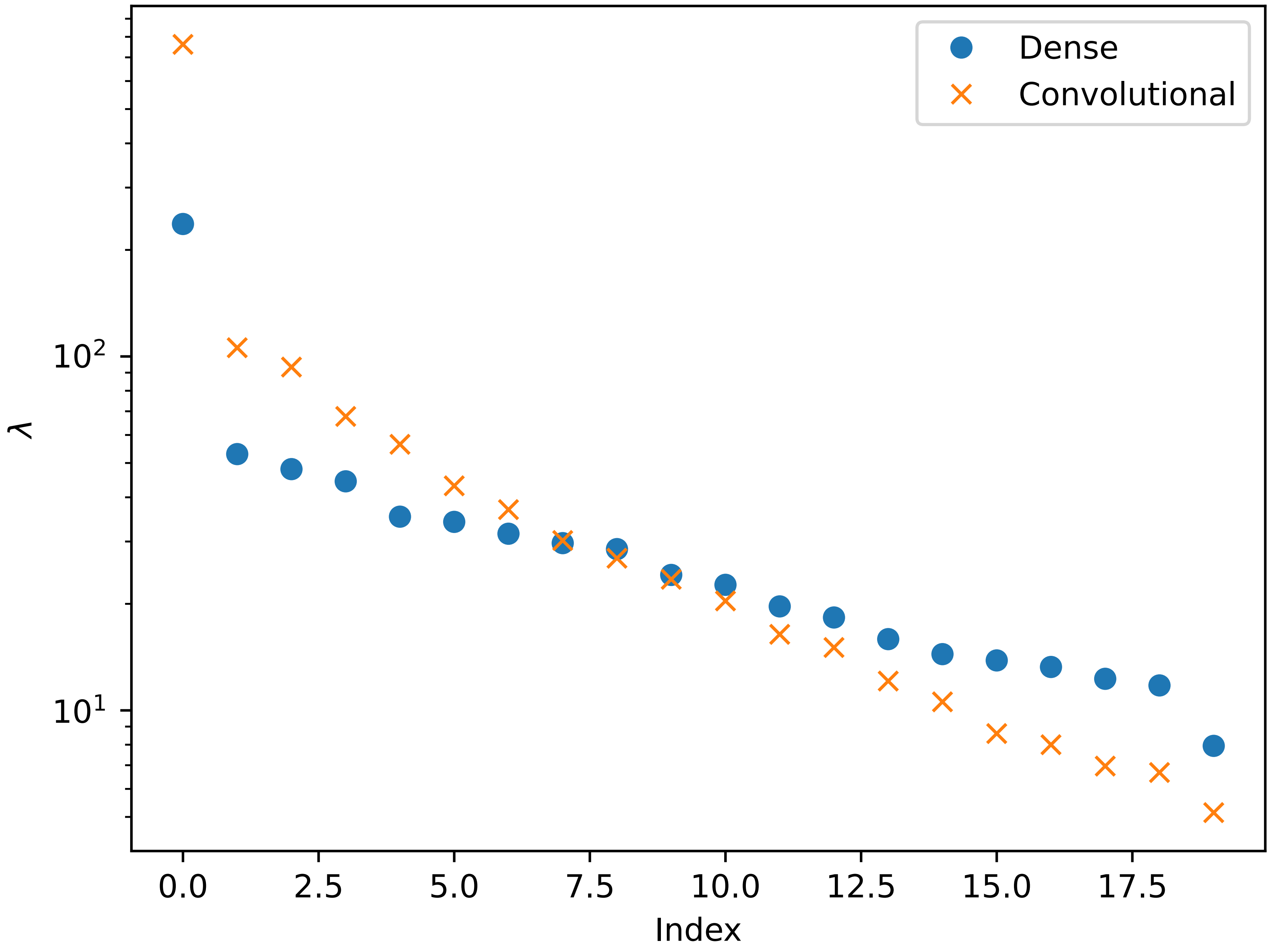
- We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

$$\mathrm{Tr}(\Theta_{\mathrm{NTK}}) = \sum_i \lambda_i \approx \lambda_{\max}$$

- The # of relevant modes differs between tasks. A variable which is independent of the # of modes is the following entropy:

$$S^{VN} = -\sum_i \hat{\lambda}_i \log \hat{\lambda}_i$$

(here: $\hat{\lambda}_i$ normalised eigenvalues of Θ_{NTK})



Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

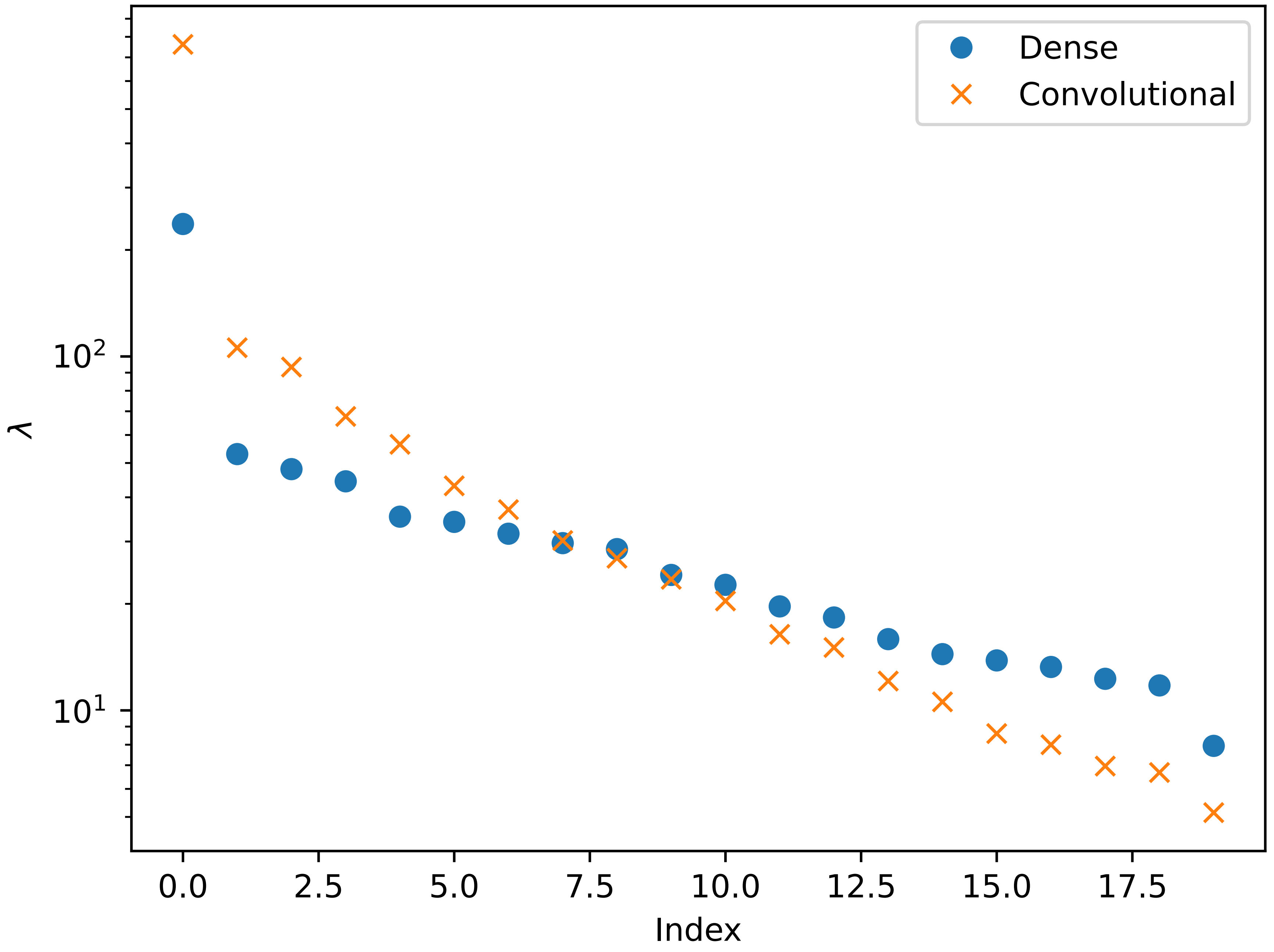
- We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

$$\mathrm{Tr}(\Theta_{\mathrm{NTK}}) = \sum_i \lambda_i \approx \lambda_{\max}$$

- The # of relevant modes differs between tasks. A variable which is independent of the # of modes is the following entropy:

$$S^{VN} = -\sum_i \hat{\lambda}_i \log \hat{\lambda}_i$$

(here: $\hat{\lambda}_i$ normalised eigenvalues of Θ_{NTK})



▸ **Which behaviour do these two variables show in neural networks?**

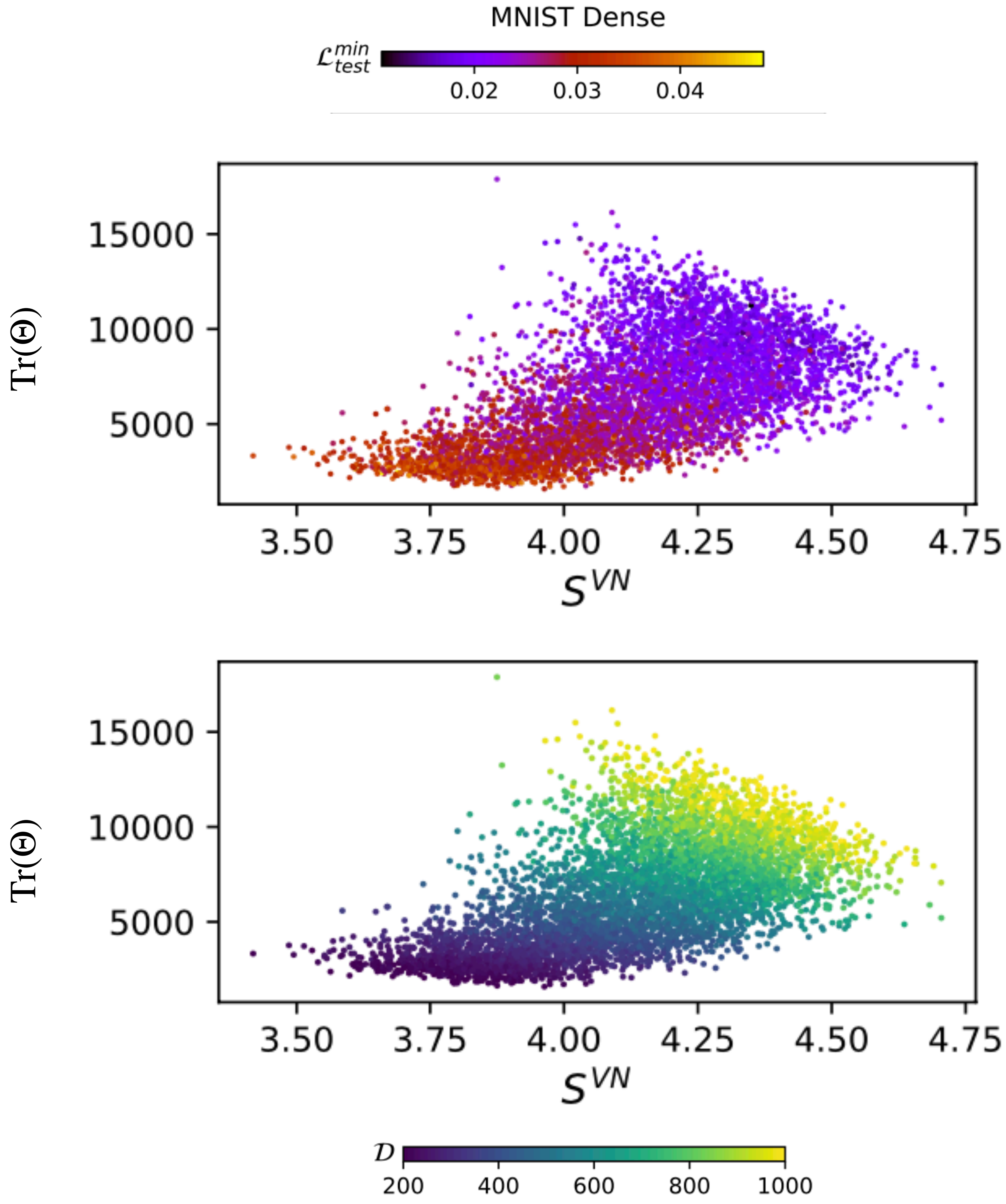
Collective variables

$\mathrm{Tr}(\Theta_{\mathrm{NTK}}), S = -\sum \lambda_i \log \lambda_i$

- Data selection changes the spectrum and sub-sequently our collective variables:

$$\dot{f}(x) = \nabla_\theta f(x)\ \dot{\theta} = -\eta \Theta(x,y) \nabla_{f(y)} \mathscr{L}$$

- We see that both collective variables (trace and entropy of empirical NTK) correlate with generalisation behaviour.
- These collective variables seem like an interesting window into data selection.

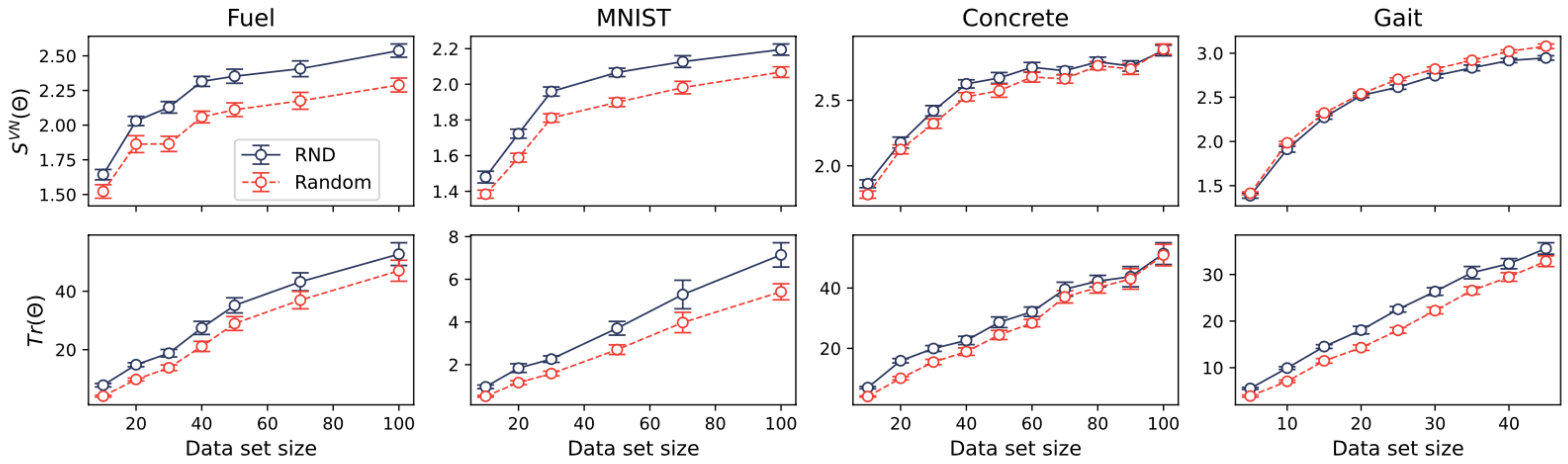


Disentangling: data size and collective variables

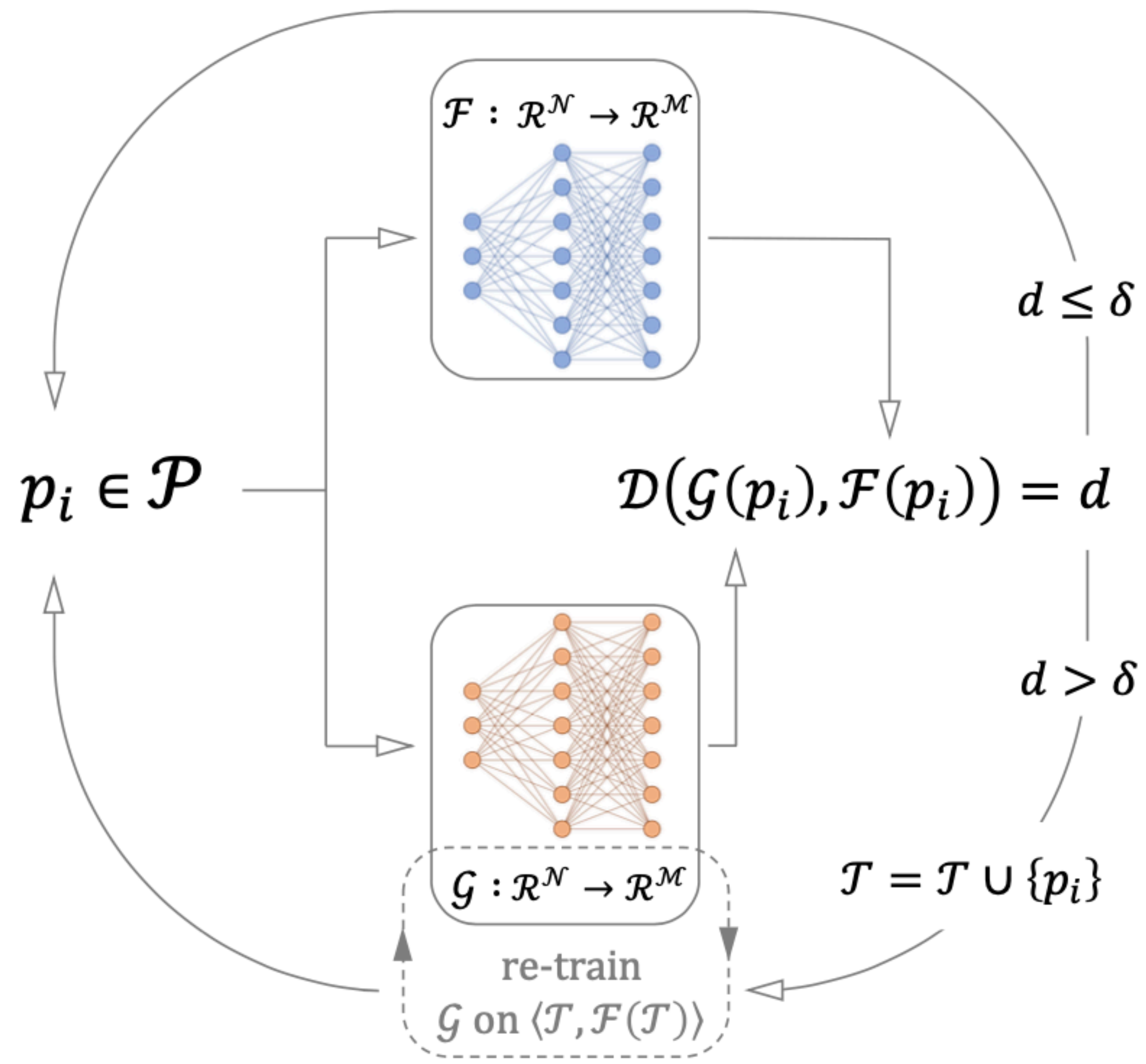
Data selection: Random Network Distillation (RND)

- What happens when we select datasets of the same size but with different collective variables values?
- We use Random Network Distillation (Burda et al. 2018) and randomly selected samples.
- RND selected samples show larger collective variables.

Different datasets



Workflow of RND

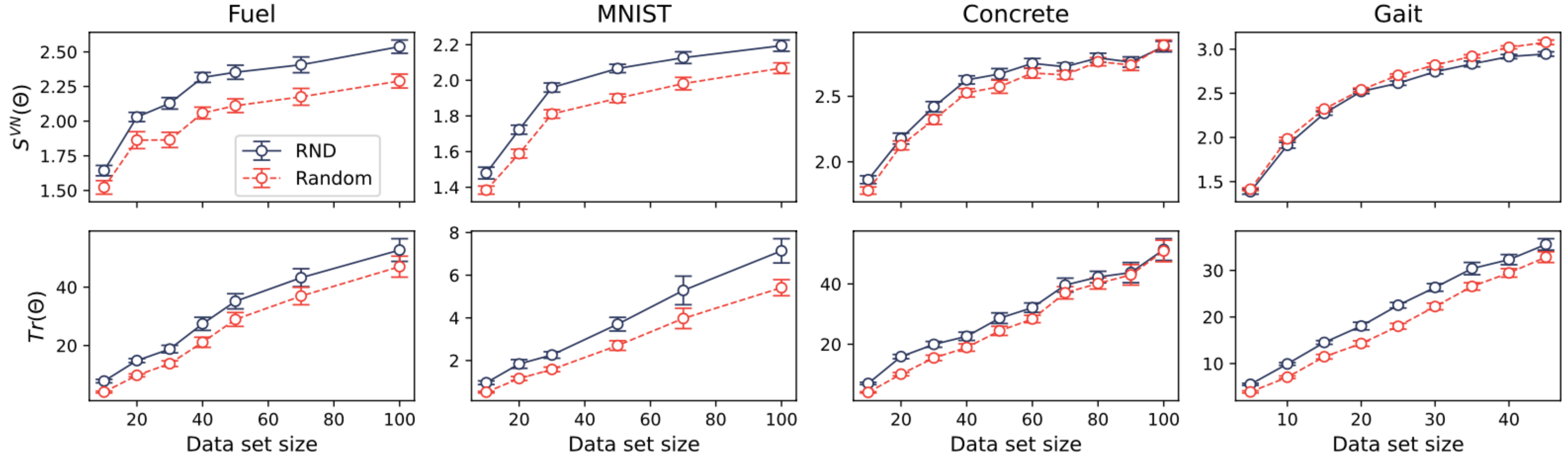


Disentangling: data size and collective variables

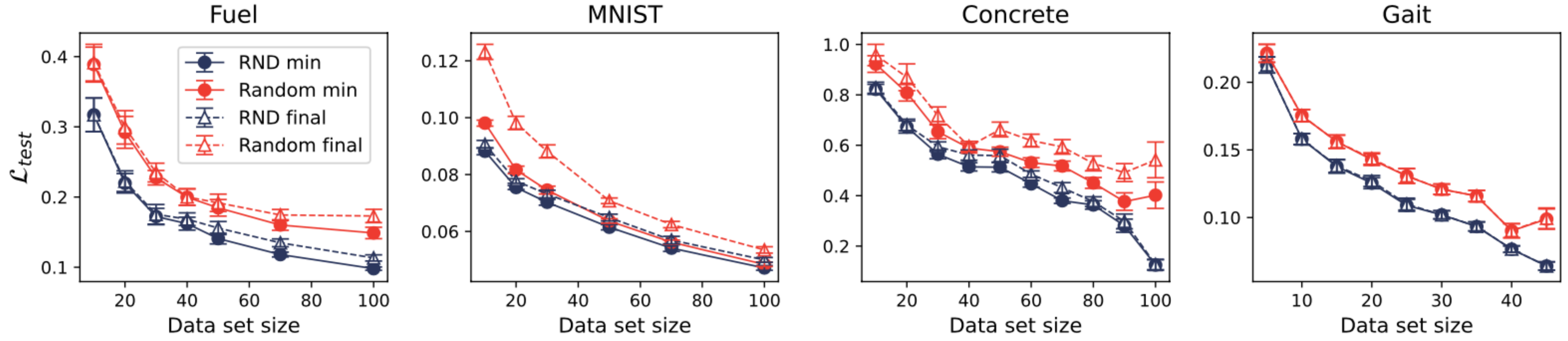
Data selection: Random Network Distillation (RND)

- RND datasets: higher collective variables and better test performance. **NNs can be more efficient with those datasets.**

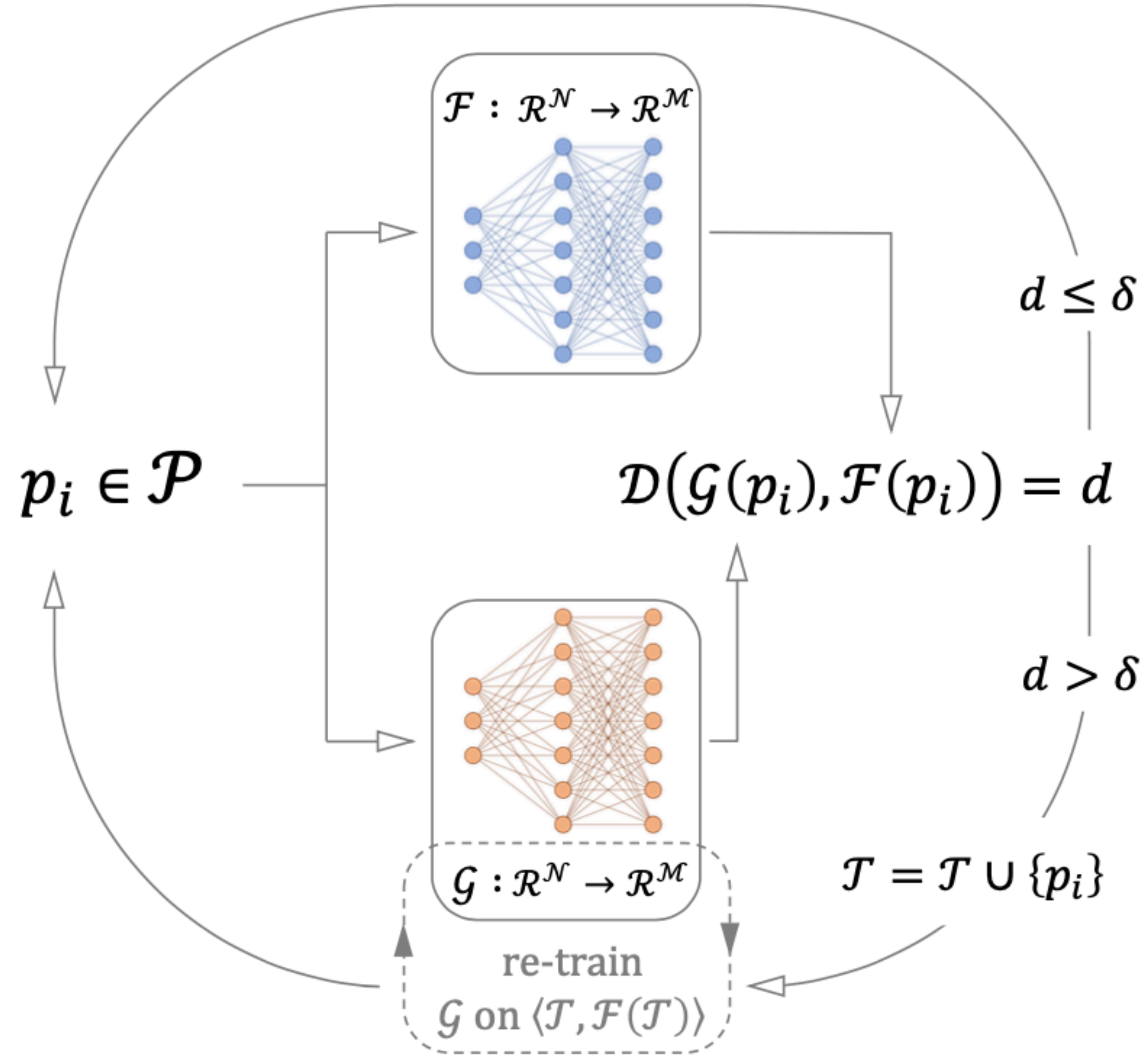
Different datasets



Test performances



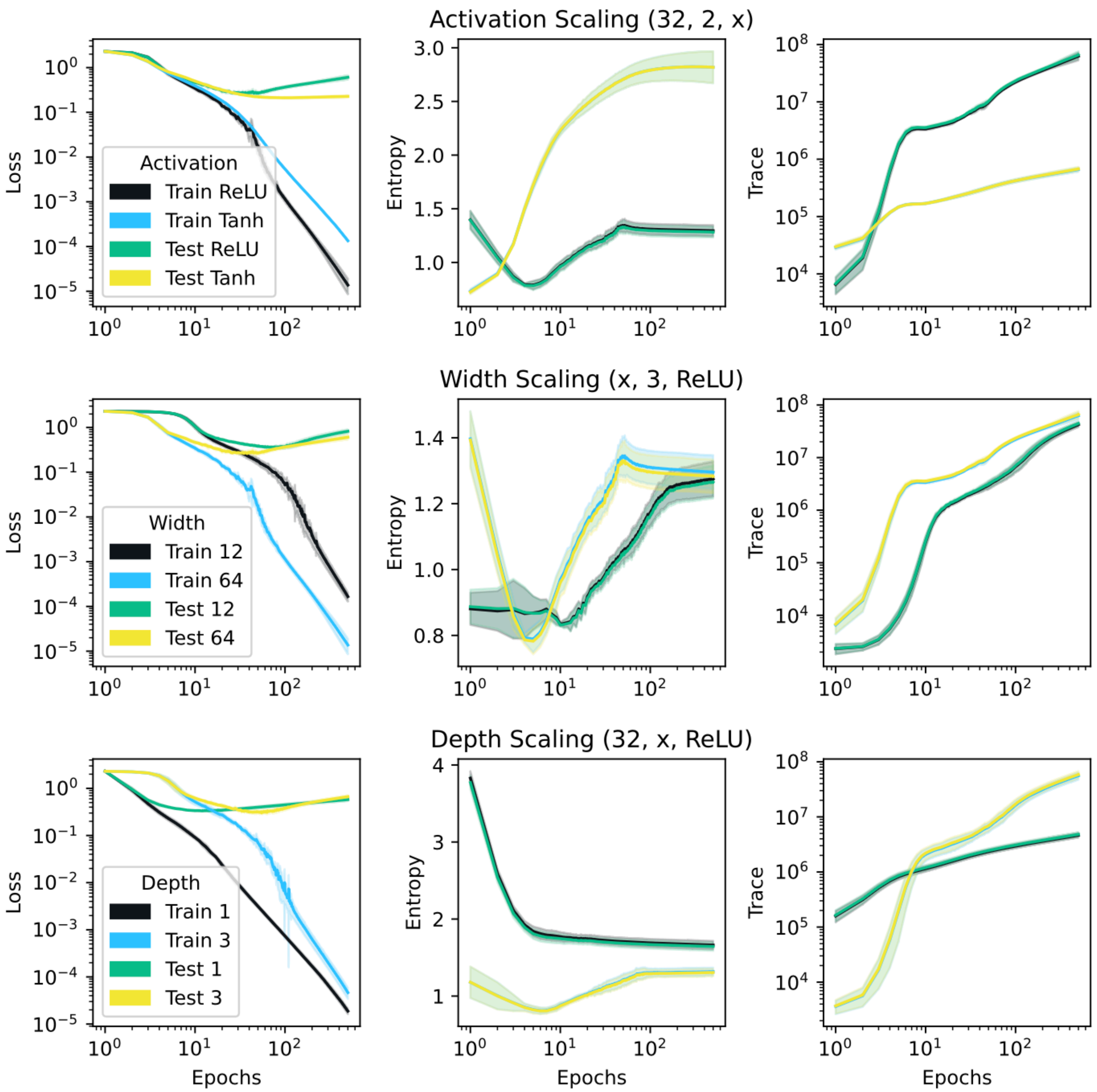
Workflow of RND



How do collective variables evolve during training?

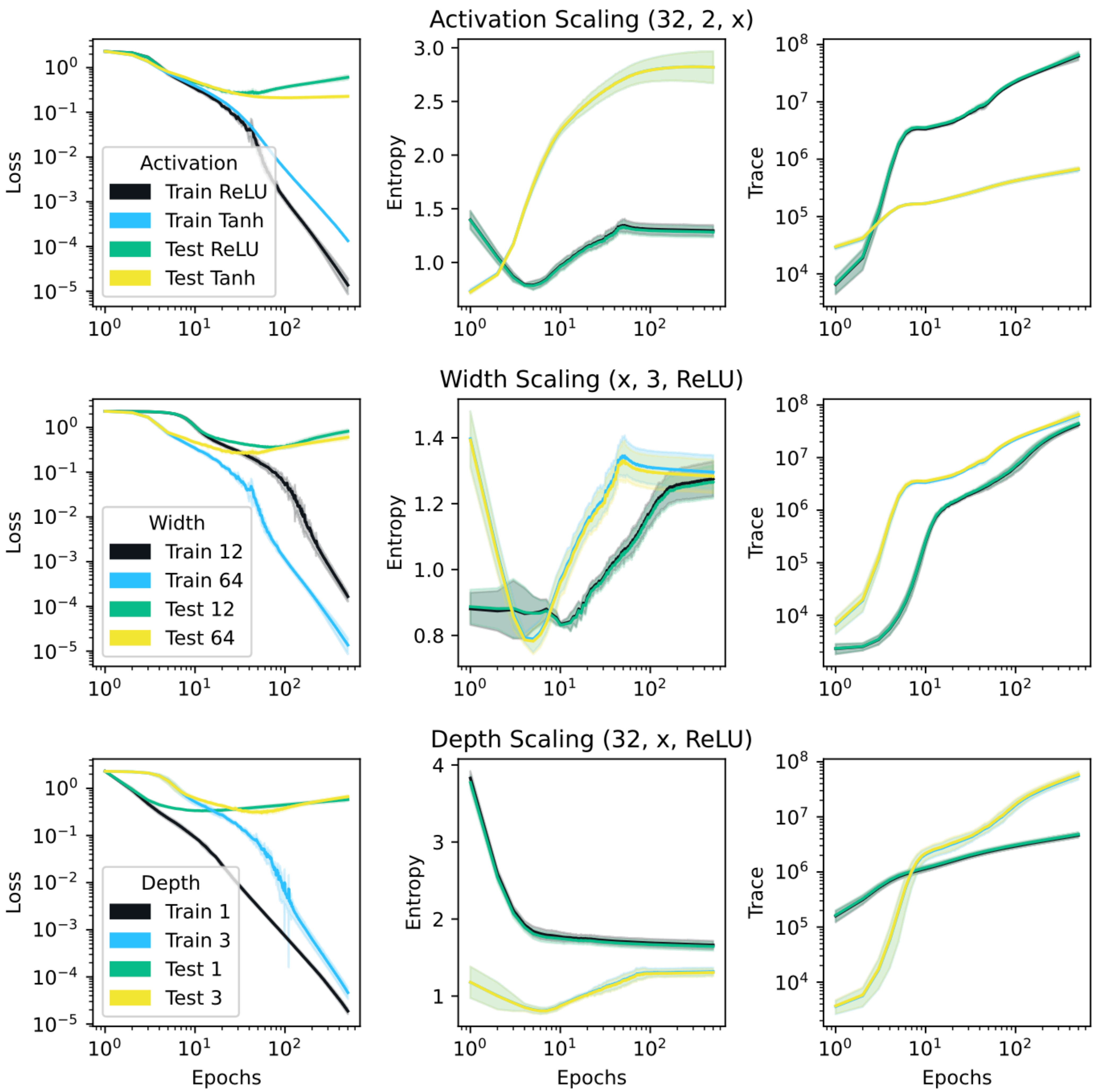
2410.07451

Collective variable evolution



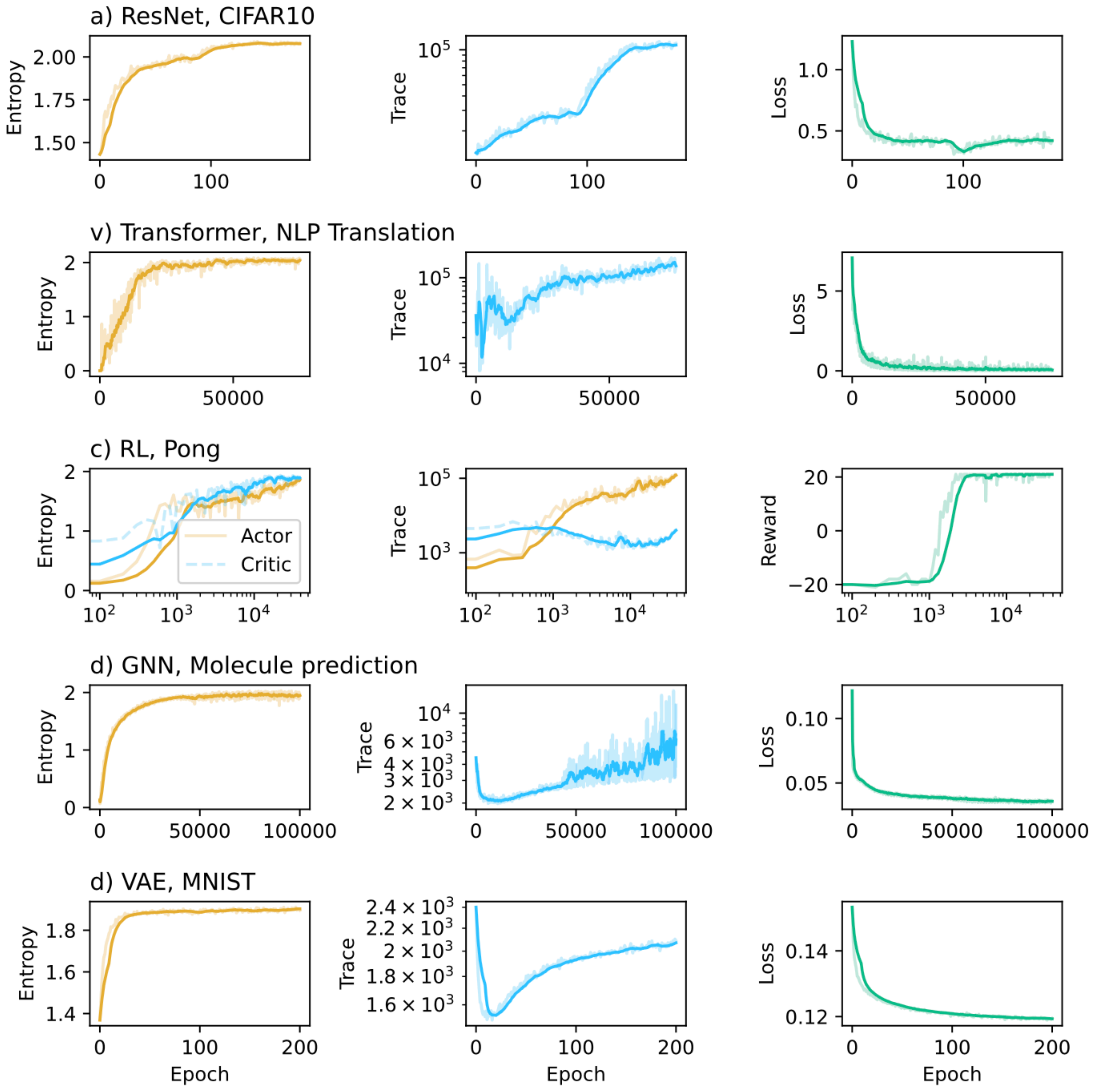
Collective variable evolution

- Entropy: two distinctive learning phases (first going down [compression], then going up for larger networks [structure formation])
- Trace: increasing throughout learning (effective learning rate)



Collective variable evolution

- Entropy: two distinctive learning phases (first going down [compression], then going up for larger networks [structure formation])
- Trace: increasing throughout learning (effective learning rate)
- Behaviour also seen in novelty models



Understand NTK dynamics by matching with (non-linear) physical systems…

Understand NTK dynamics by matching with (non-linear) physical systems…

I find it easier when I have a second order differential equation.

The career of a young theoretical physicist consists of treating harmonic oscillator in ever-increasing levels of abstraction.
Sidney Coleman

Neural Networks

Gradient descent with momentum

- Modification of optimizer: gradient descent with momentum

$$\theta_{i+1} = \theta_i + v_i \ , \quad v_i = \beta v_{i-1} - \eta \nabla_\theta \mathscr{L}$$

- NN differential equation becomes second order (more familiar from scalar field dynamics)

$$\ddot{f}(x) + \frac{1-\beta}{\sqrt{\eta}} \dot{f}(x) + \Theta(x,y)\mathscr{L}'(f(y)) = 0 \qquad (\Delta t = i\sqrt{\eta})$$

let's return to this system in a second…

Let’s look at some physical system:

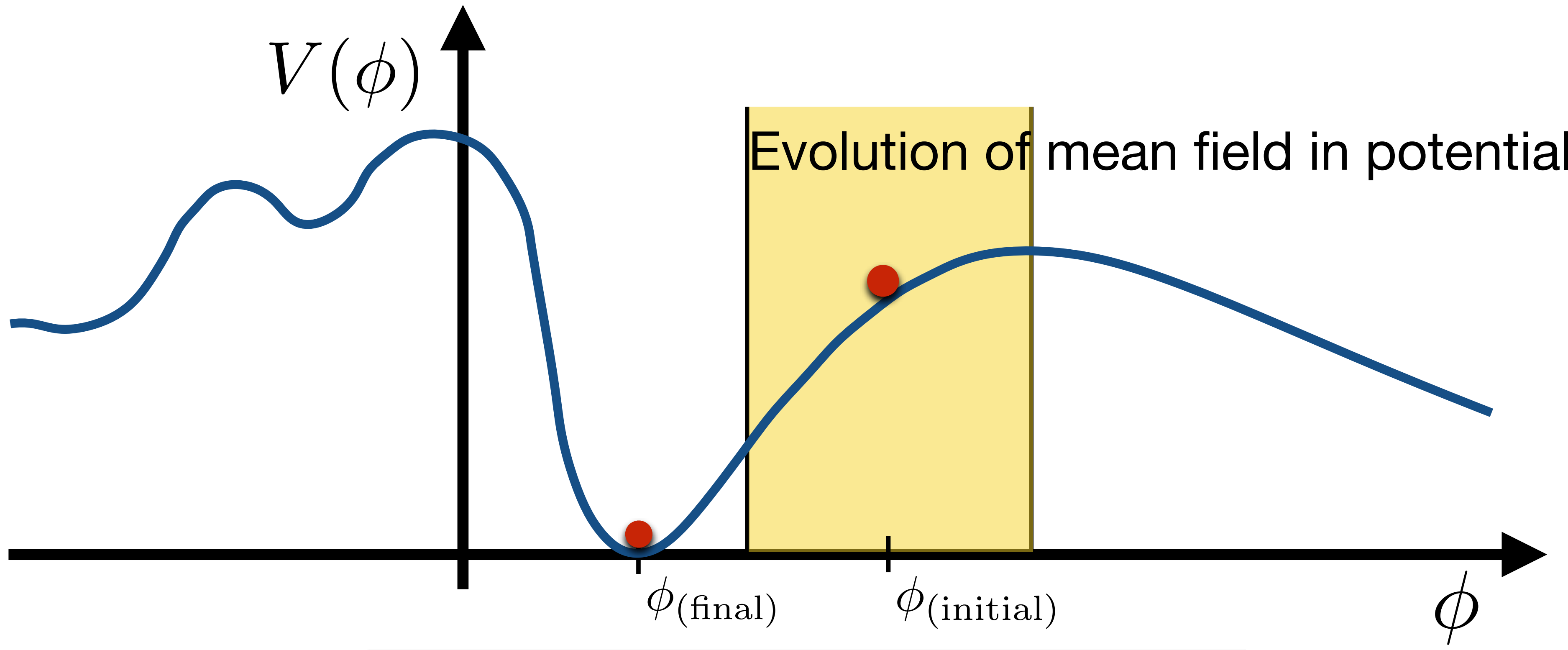
Scalar fields in FLRW

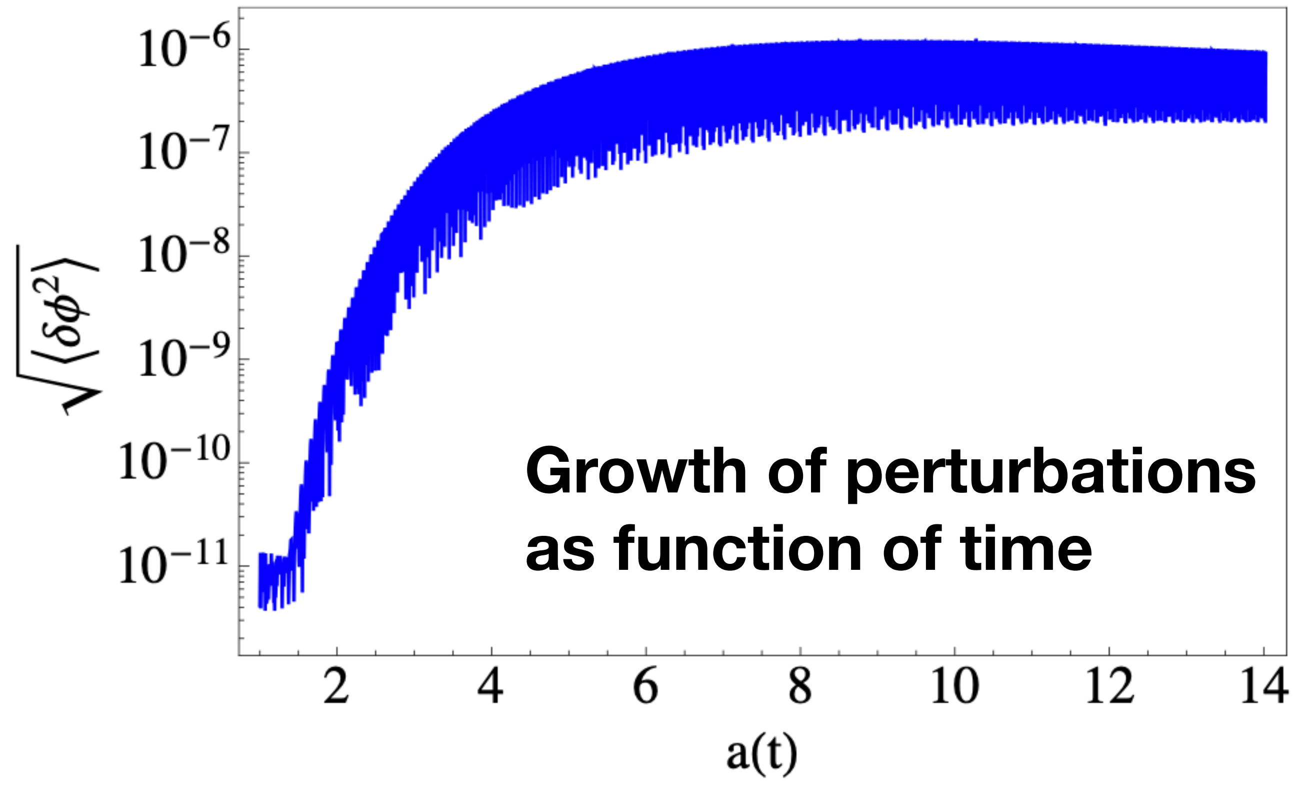
Cosmological toy models

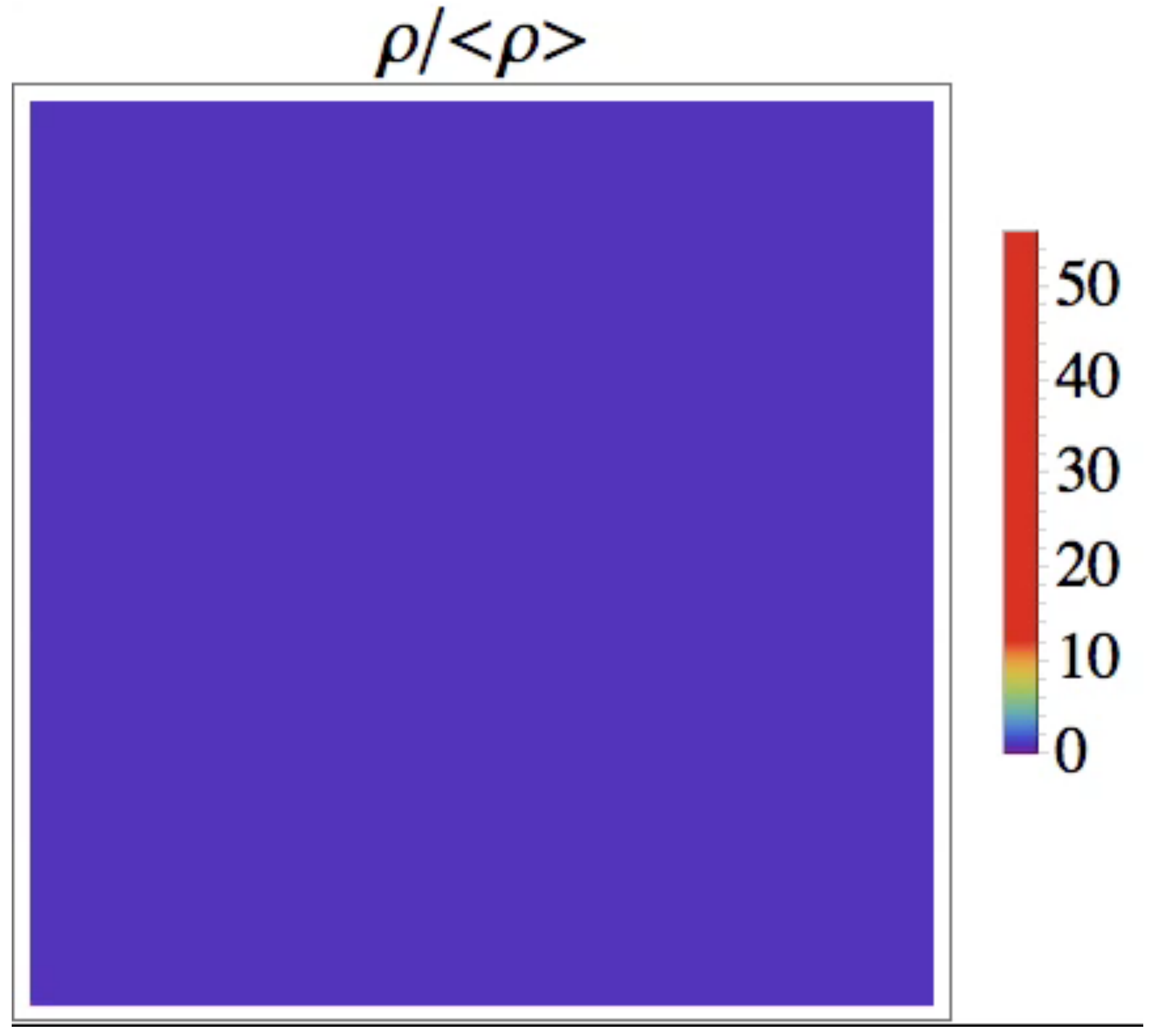
- Scalar field in FLRW
$$S = \int d^4x\sqrt{-g}\left(R + \frac{1}{2}g^{\mu\nu}\partial_\mu\phi\partial_\nu\phi - V(\phi)\right)$$
- Homogeneous scalar field $\phi(x,t) = \phi(t)$ eom:
$$\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0\ ,\quad 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi)$$
- Fluctuations around homogeneous background
$\phi(x,t) = \phi(t) + \delta\varphi(x,t)$:
$$\ddot{\varphi} + \nabla^2\varphi + 3H\dot{\varphi} + V''(\phi)\varphi = 0$$

Dynamical properties:

- Perturbations (≈ non-trivial features) can grow and remain.
- Modes can (temporarily) freeze during cosmological evolution.

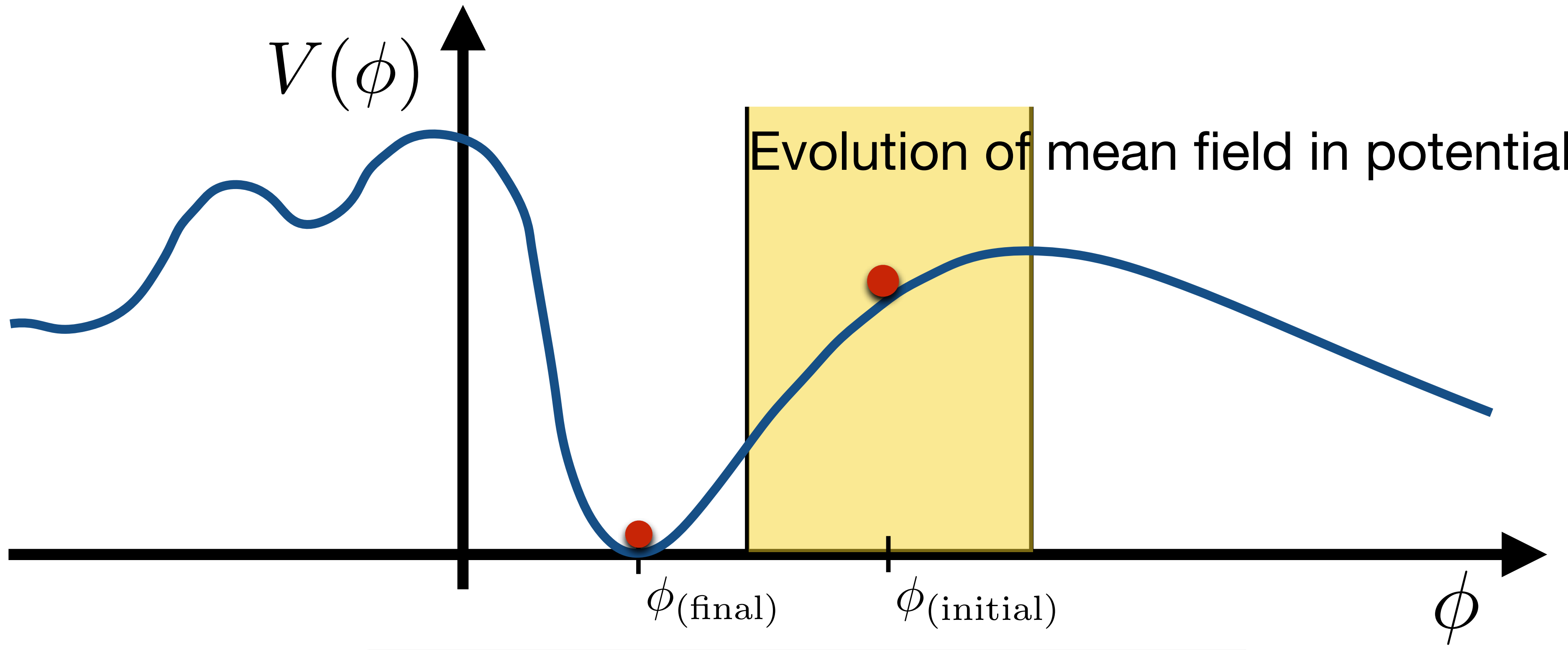




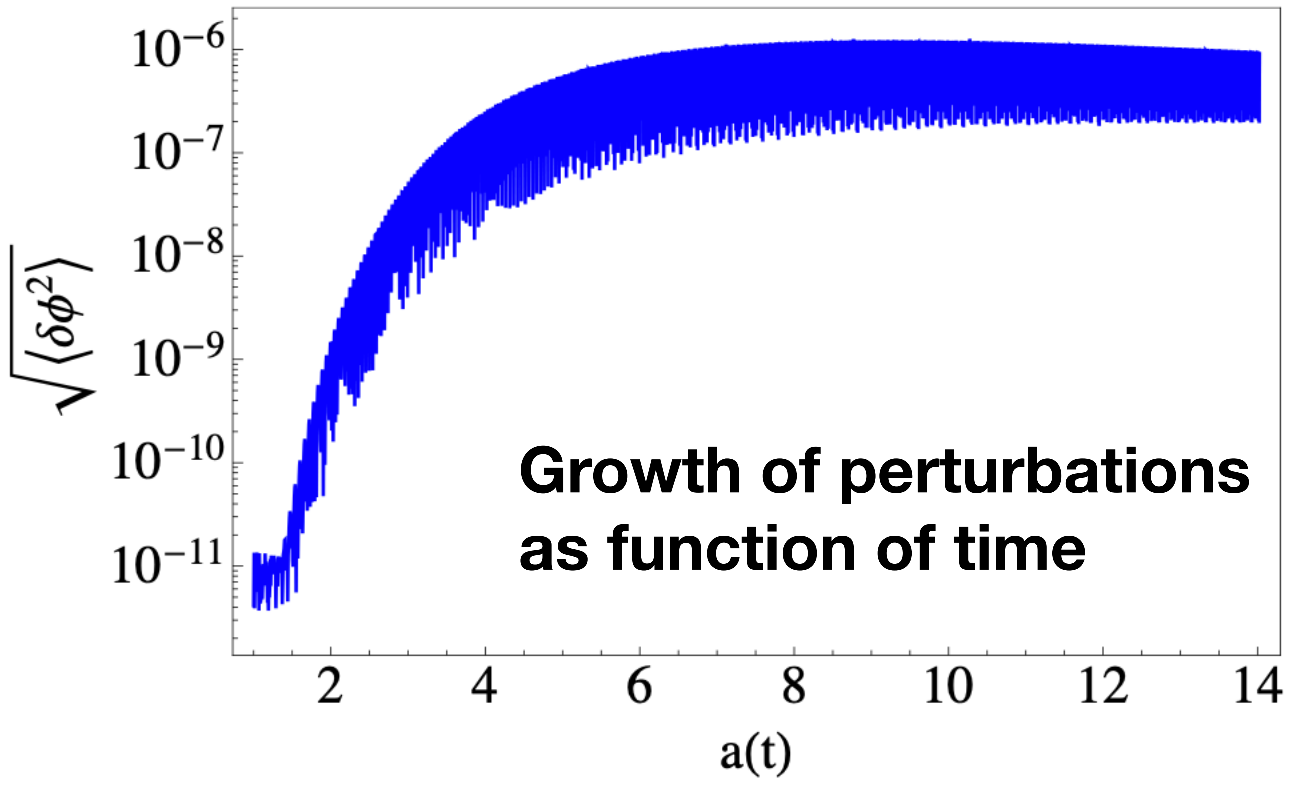


Antusch, Cefala, Krippendorf, Muia, Orani, Quevedo 2016

Cosmological toy models

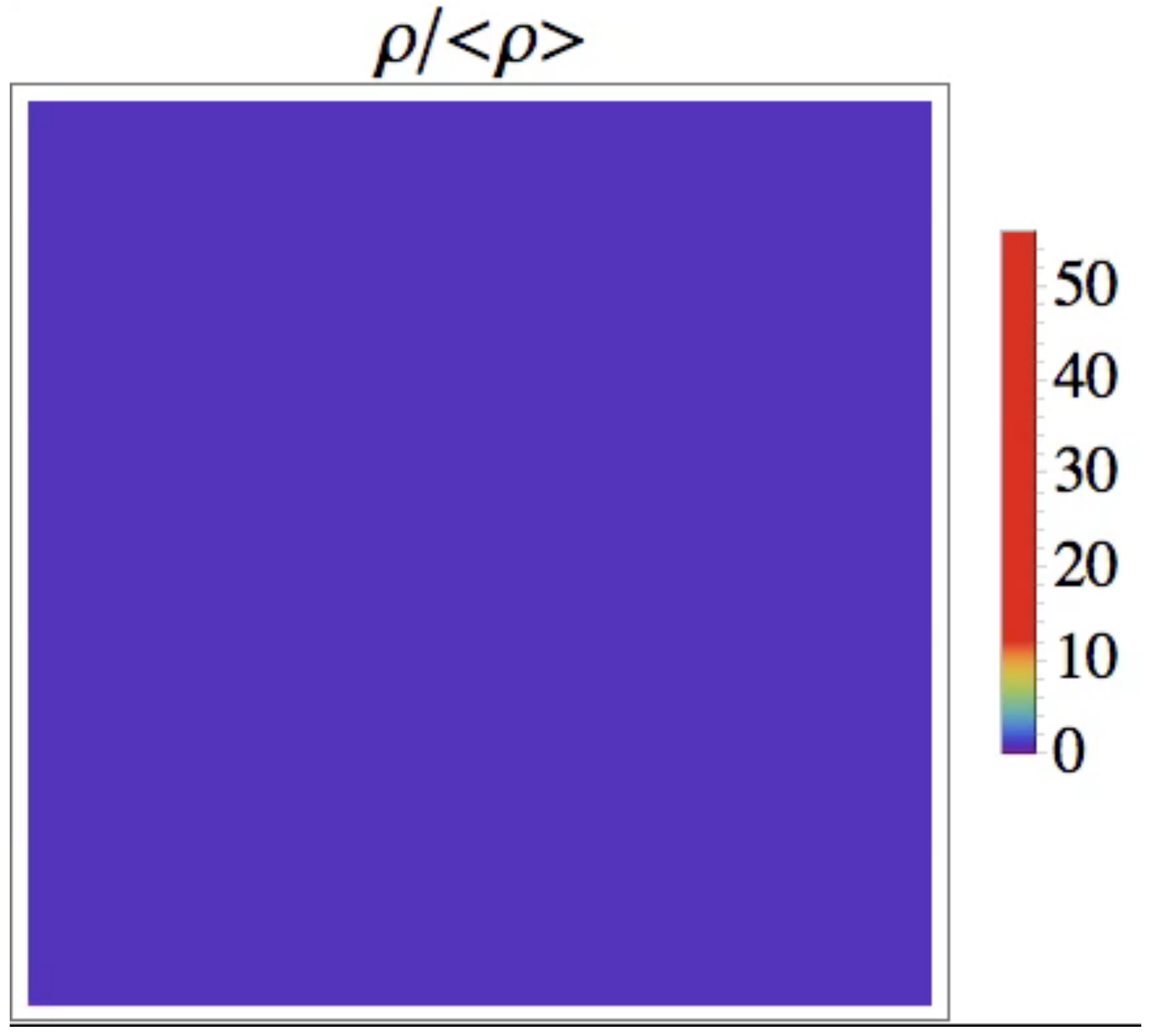


- Scalar field in FLRW
$$S = \int d^4x\sqrt{-g}\left(R + \frac{1}{2}g^{\mu\nu}\partial_\mu\phi\partial_\nu\phi - V(\phi)\right)$$
- Homogeneous scalar field $\phi(x,t) = \phi(t)$ eom:
$$\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0\ ,\quad 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi)$$
- Fluctuations around homogeneous background
$\phi(x,t) = \phi(t) + \delta\varphi(x,t)$:
$$\ddot{\varphi} + \nabla^2\varphi + 3H\dot{\varphi} + V''(\phi)\varphi = 0$$



Dynamical properties:

- Perturbations (≈ non-trivial features) can grow and remain.
- Modes can (temporarily) freeze during cosmological evolution.



Antusch, Cefala, Krippendorf, Muia, Orani, Quevedo 2016

Are these two dynamical systems comparable?

Are these two dynamical systems related?

Homogeneous and isotropic dynamics

- NN eqn:

$$\ddot{f}(x) + \frac{1-\beta}{\sqrt{\eta}}\dot{f}(x) + \Theta(x,y)\mathscr{L}'(f(y)) = 0$$

- Homogeneous NN $f(x,t) = f(t)$: no input dependence $\rightarrow \quad \Theta(x,y) = \alpha$

$$\ddot{f} + \frac{1-\beta}{\sqrt{\eta}}\dot{f} + \alpha\mathscr{L}'(f) = 0$$

- Scalar field:

$$\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0\ , \quad 3H^2 = \frac{\dot{\phi}^2}{2} + V(\phi)$$

- Vacuum energy dominated Universe $H \approx \sqrt{V_0}/3 = \text{const.}$:

$$\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0$$

$$V_{\text{eff}} = \alpha\mathscr{L} + V_0\ , \ V_0 = \frac{\tilde{\beta}^2}{3}$$

Are these two dynamical systems related?

Perturbation dynamics

- NN evolution equation:

$$\ddot{f}(x) + \tilde{\beta}\dot{f} + \Theta(t, x, X)\,\nabla_{f(X)}\mathscr{L} = 0$$

- Rewrite last-term:

$$\Theta(t, x, X)\,\nabla_{f(X)}\mathscr{L} = \sum_y \Theta(t, x, y)\mathscr{L}'(f(y)) = \sum_y \Theta(t, x, y)(\mathscr{L}'(\bar{f}) + \mathscr{L}''(\bar{f})\delta f(y))$$

- Basis transformation to diagonalise NTK-kernel:

$$A\Theta A^T = \mathrm{diag}(\lambda_1, \ldots, \lambda_N),\, A = \begin{pmatrix} v_1^T \\ v_2^T \\ \cdots \\ v_N^T \end{pmatrix},\; f_i = \sqrt{N} v_1 \bar{f} + \delta f_i \;,\; \text{and } \mathscr{L}_i = \frac{m^2}{N}(f_i - f_0)$$

- Rewriting equations $\delta\tilde{f}_i = v_i^T \cdot \delta\mathbf{f}$:

$$0 = \ddot{\bar{f}} + \tilde{\beta}\,\dot{\bar{f}} + \frac{\lambda_1}{N}\,\mathscr{L}'(\bar{f}) \;,\quad 0 = \delta\ddot{\tilde{f}}_i + \tilde{\beta}\,\delta\dot{\tilde{f}}_i + \frac{\lambda_i}{N}\,\delta\tilde{f}_i\,\mathscr{L}''(\bar{f})$$

Are these two systems related? YES

- Exact match subject to the following assumptions: vacuum energy dominated universe, low momentum modes ($\nabla^2\varphi \approx 0$), $\lambda_i \approx \lambda_1$

$$0 = \ddot{\bar{f}} + \tilde{\beta}\,\dot{\bar{f}} + \frac{\lambda_1}{N}\,\mathscr{L}'(\bar{f})\ ,\quad 0 = \delta\ddot{\tilde{f}}_i + \tilde{\beta}\,\delta\dot{\tilde{f}}_i + \frac{\lambda_i}{N}\,\delta\tilde{f}_i\,\mathscr{L}''(\bar{f})$$

$$\ddot{\phi} + 3H\dot{\phi} + V'(\phi) = 0\ ,\quad \ddot{\varphi} + \nabla^2\varphi + 3H\dot{\varphi} + V''(\phi)\varphi = 0$$

- Remarkable: structurally the equations look very similar even when relaxing conditions (e.g. $\lambda_i \approx \lambda_1$)

- Hubble scale set by optimiser parameters:

$$3H = \frac{1-\beta}{\sqrt{\eta}}$$

- Adding more data points = adding more modes in evolution. If modes are irrelevant, they can be neglected. Understand generalisation behaviour for finite data?

Quantitatively matching dynamics

Quantitatively matching dynamics

Scope

1. When are these equations capturing the NN dynamics accurately?

$$0 = \ddot{\bar{f}} + \tilde{\beta}\,\dot{\bar{f}} + \frac{\lambda_1}{N}\,\mathscr{L}'(\bar{f})\;, \quad 0 = \delta\ddot{\tilde{f}}_i + \tilde{\beta}\,\delta\dot{\tilde{f}}_i + \frac{\lambda_i}{N}\,\delta\tilde{f}_i\,\mathscr{L}''(\bar{f})$$

2. How do the EFT parameters depend on the NN parameters, e.g. determining α empirically.

- Utilising existing empirical NTK implementation

https://github.com/google/neural-tangents

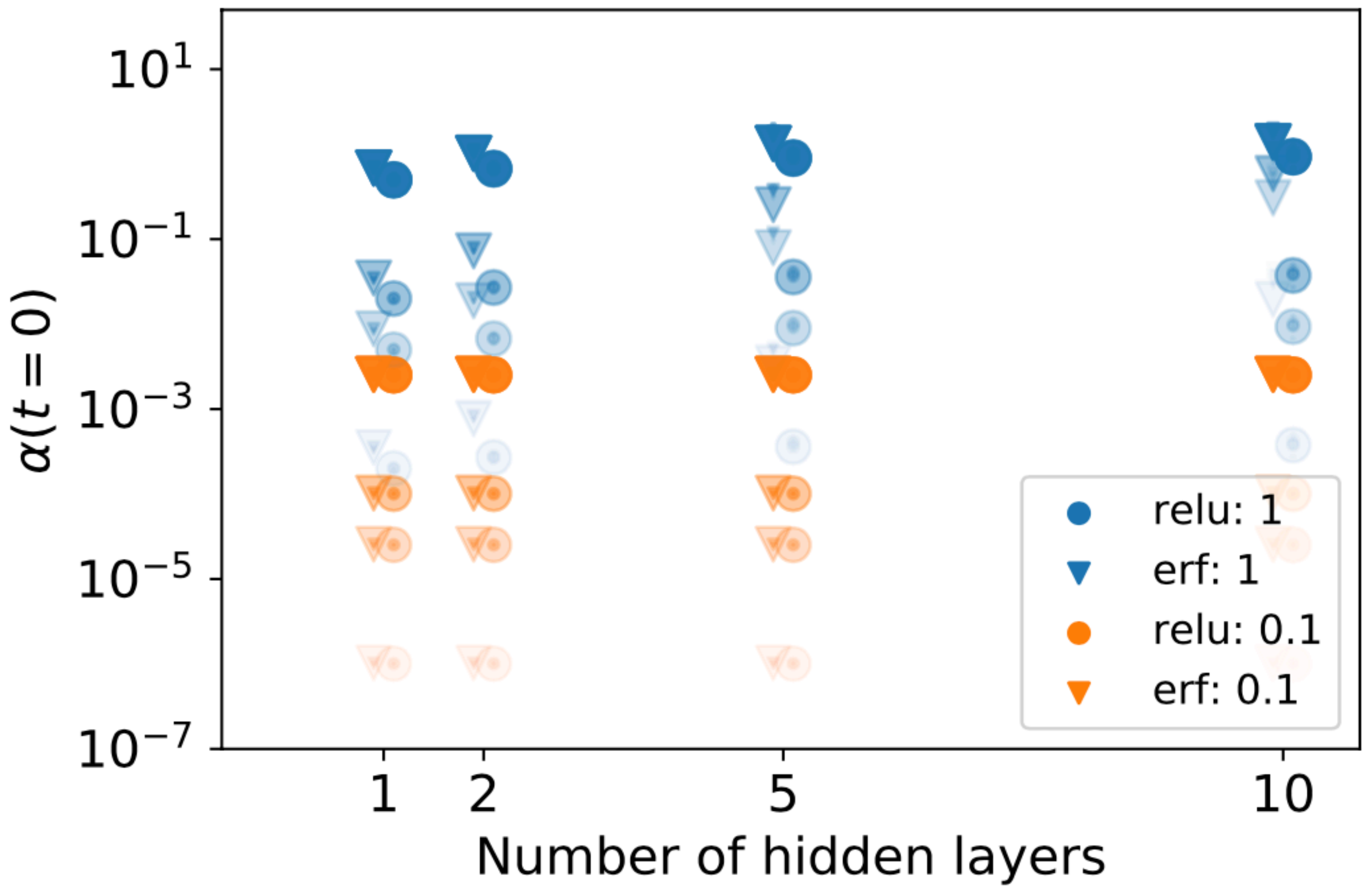
Quantitatively matching dynamics

NTK contribution to EFT potential

- How does the NTK vary when changing hyperparameters:

$$V_{\text{eff}} = \alpha \mathscr{L} + V_0$$
$$\mathscr{L} = \frac{m^2}{2}(f - f_0)^2$$

- Initialisation is most sensitive, scales in α vary largely



Quantitatively matching dynamics

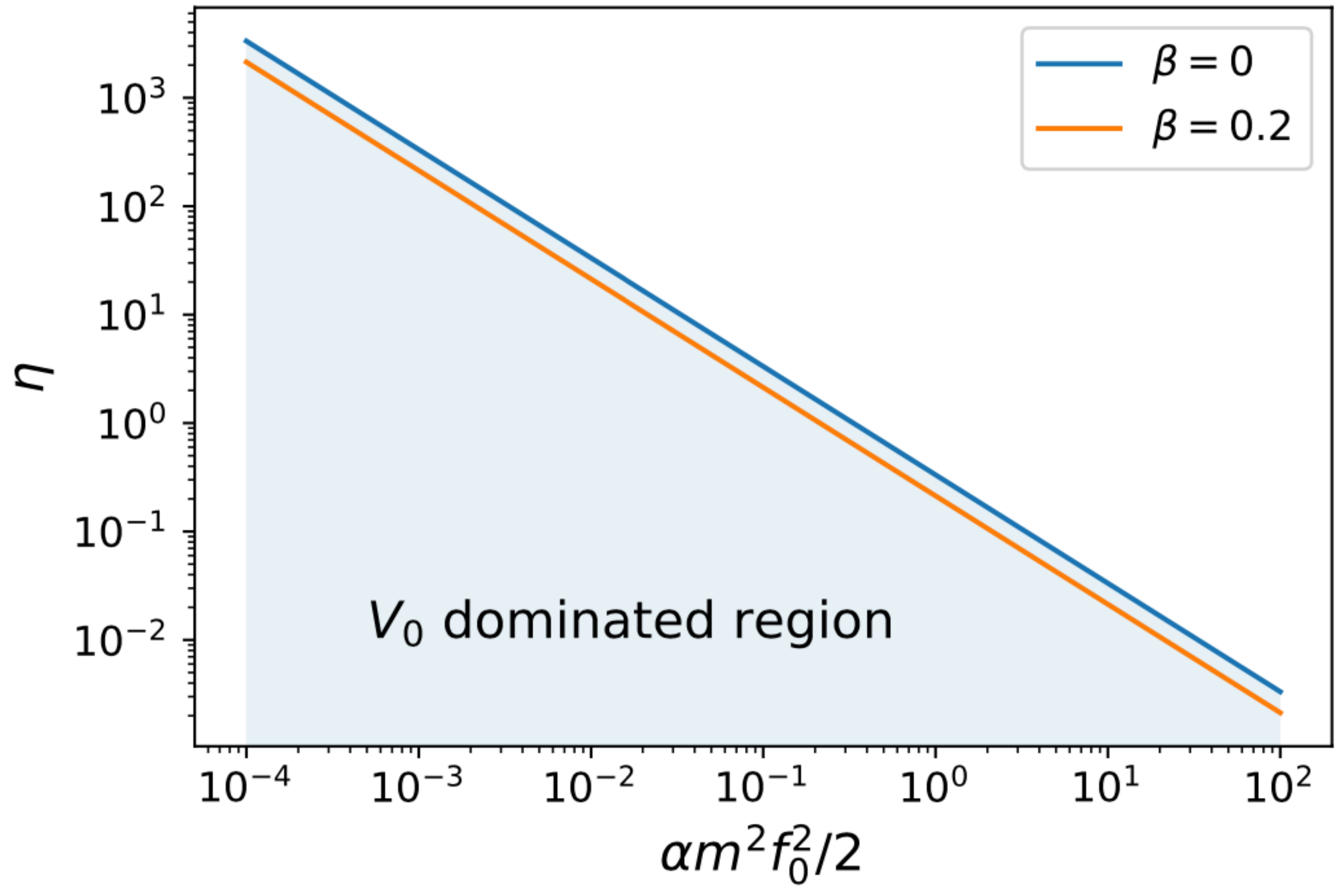
When are we in a vacuum energy dominated regime?

- Vacuum energy needs to dominate over other contributions to Hubble:

$$3H = \frac{1-\beta}{\sqrt{\eta}}$$

$$\mathscr{L} = \frac{m^2}{2}(f - f_0)^2$$

- Reasonable learning rates are allowed!



Quantitatively matching dynamics

Homogeneous case

- Train homogeneous NNs with different hyperparameters and solve differential equation to check predicted vs. actual gradient descent evolution:

$$\ddot{f} + \frac{\beta - 1}{\sqrt{\eta}}\dot{f} + \alpha \mathscr{L}'(f) = 0$$

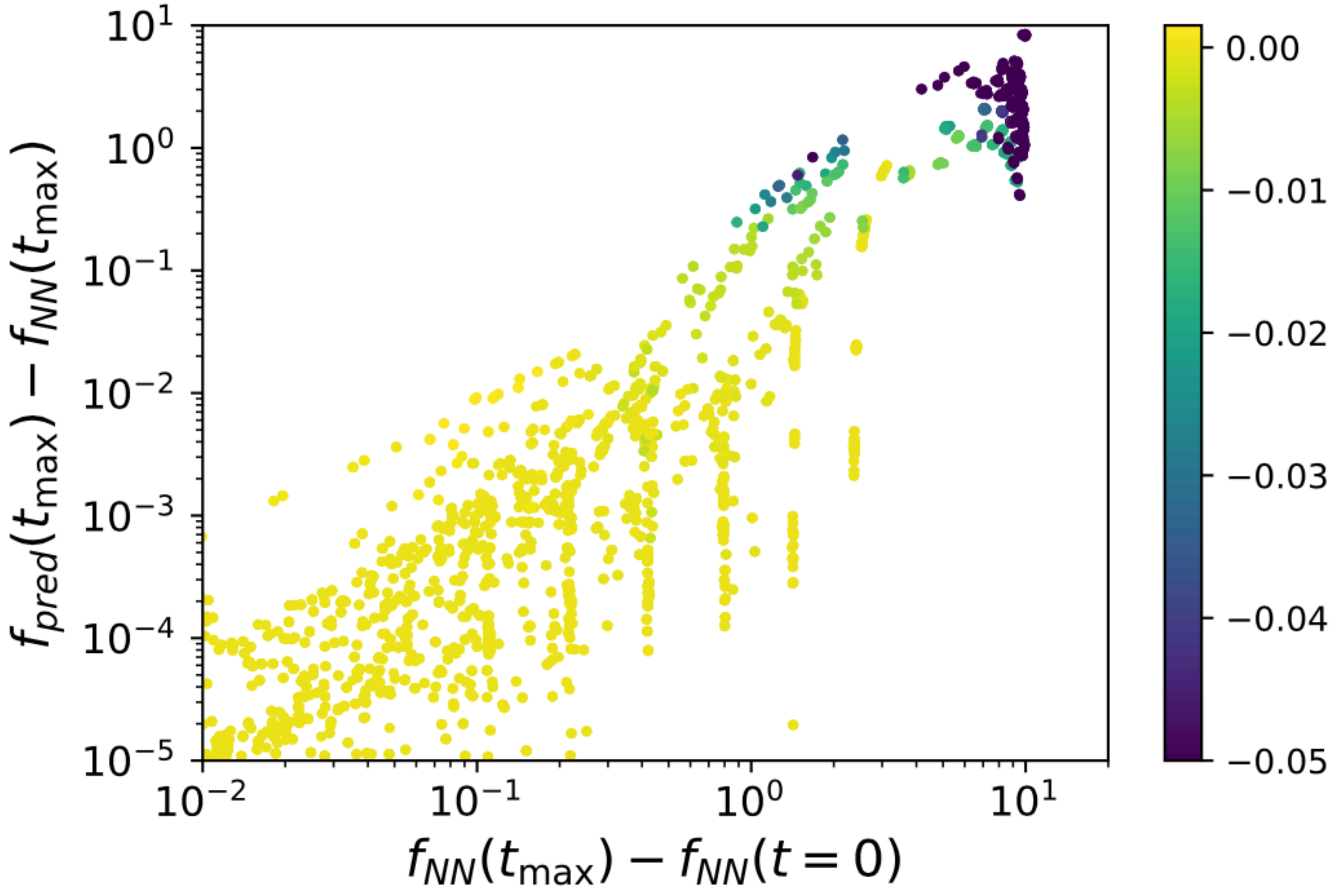
- Loss function, e.g.:

$$\frac{m^2}{2}(f - f_0)^2$$

- Colour: $\alpha(t_{\max}) - \alpha(t = 0)$

- Hyperparameters: {1,2,5} hidden layers, {erf, relu} activation, {100,1000} width, initialisations, learning and momentum rates

Comparing scale of evolution and error of evolution



What have we learned about NN dynamics?

- Understanding NN dynamics is key for efficient machine learning (avoiding costly searches and being energy efficient)
- How? Large parameter limit simplifies dynamics: neural tangent kernel $\nabla_\theta f(x) \nabla_\theta f(y)$
- Sparsity of NTK can be leveraged (e.g. select data) by looking at few collective variables (e.g. entropy, trace)
- Evolution of collective variables follows "universal" pattern. First empirical steps in enumerating patterns. How are these pattern related with learning behaviours? How can we use these behaviours for suitable design of NNs?
- The NN dynamics are closely connected to scalar field theory dynamics in a FLRW background.
- How to go beyond initial studies? Many experiments, need to scale up efforts.

“Give me string models that realise $|W_0| = 100$.”

“Work out the EFT underlying GPT models?”

“Search for more efficient learning algorithms.”

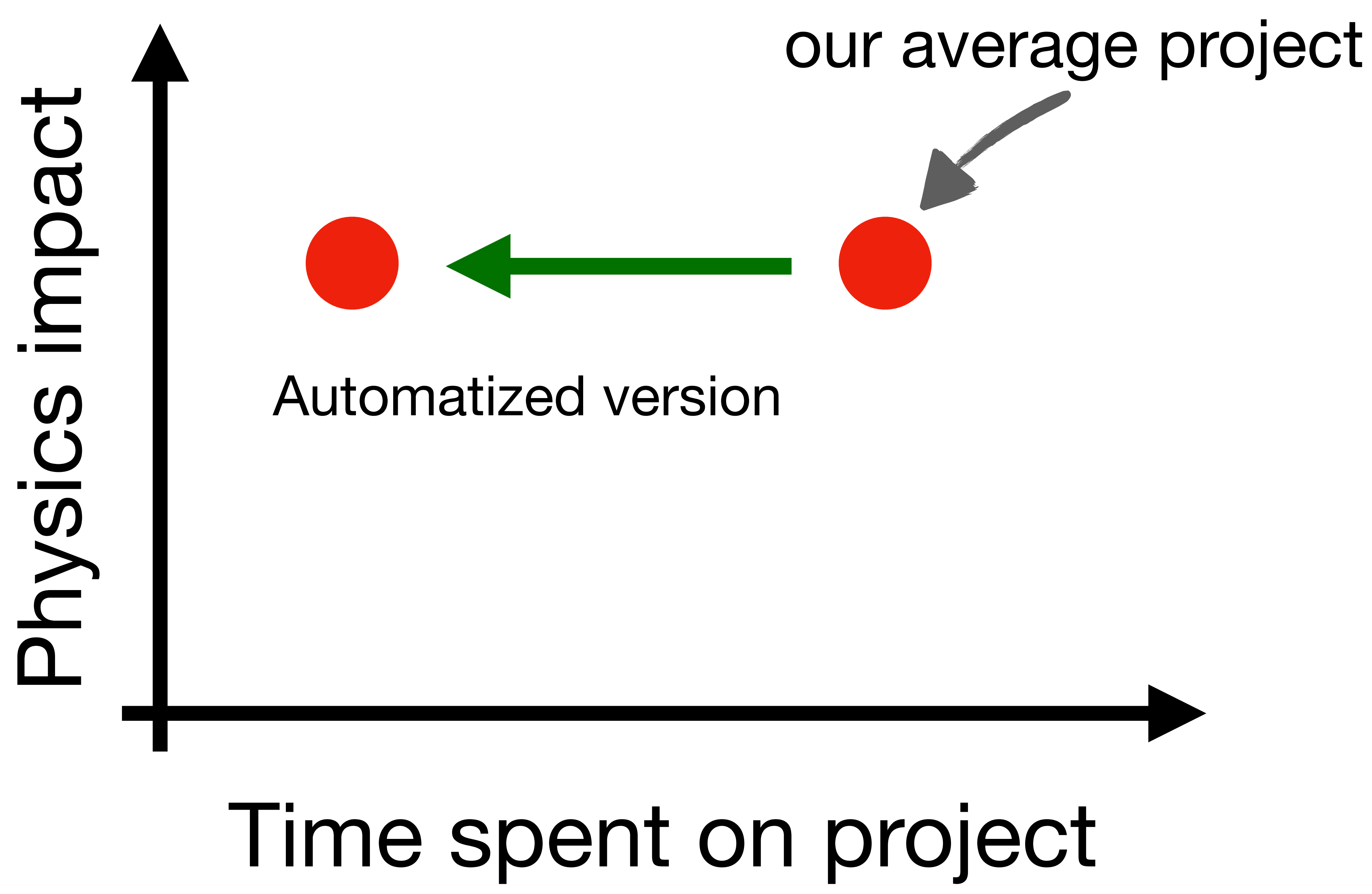
Elephant in the room: Scaling AI discoveries (physics: to other string and BSM questions; ML: automating AI research).

How can we scale up?

Automatization

What I want from automatization

- Research progress is resource limited.
- Can we use automatization to overcome resource limitation?
- Continuous improvement on efficiency leads to exponential growth in results (different scaling to scaling size or data).



Examples for automatization

Automatization → Saving time and human resources

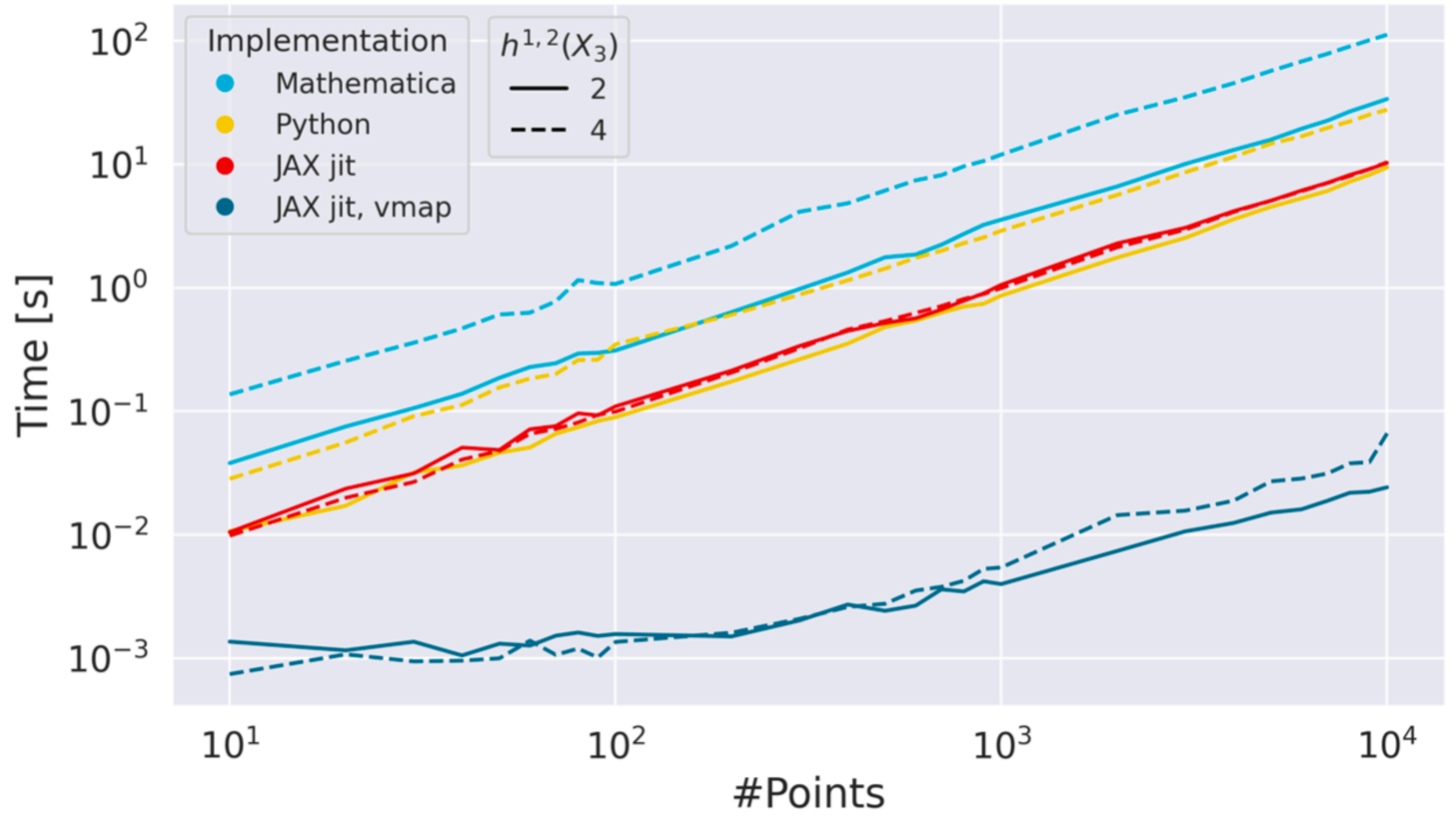
JAXVacua

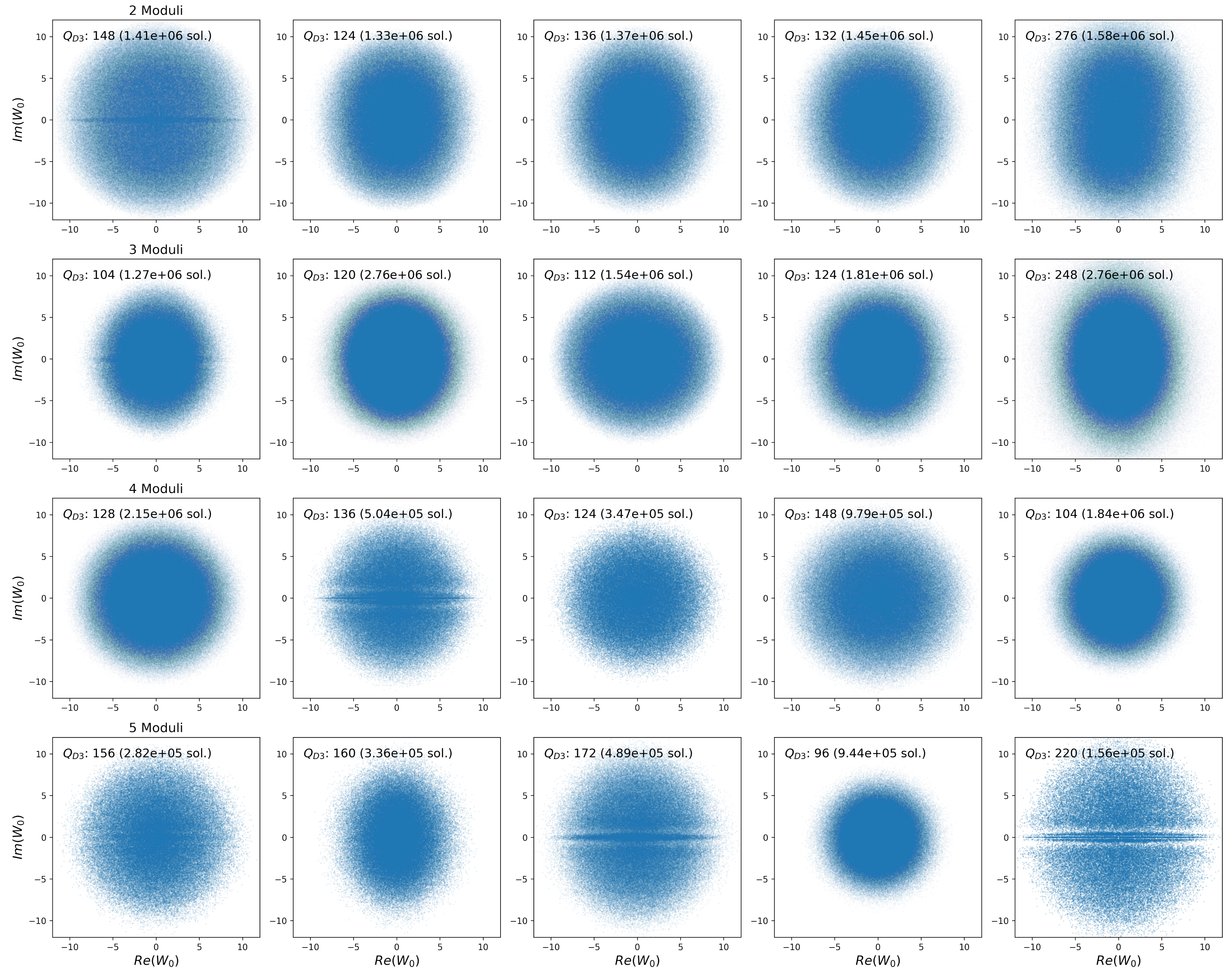
2306.06160

- **Physics Problem:** Find minima of a scalar potential arising from string theory. Potential can be calculated starting from a few topological properties. We have many compactification manifolds and many “flux” parameters, in the landscape.

- **Project prompt:** write efficient code to allow flexible study of low-energy properties such as masses of the scalar fields and allow to sample distributions of those properties.

- Results (~1 Master thesis and ~1 postdoc year later): we have solved it.





Skills

Summarizing/understanding existing results

Literature review

- Can you describe the workflow of JAXVacua (2306.06160)?
- Can you give me a technical summary of the string theory constructions used in this paper (2107.04039)? This is for experts.
- Which ML framework does this paper (1810.07703) use and which cluster properties do they study?

- # Can you describe the workflow of JAXVacua (2306.06160)?

The **workflow of JAXVacua** is modular and designed to efficiently search for string vacua using JAX's automatic differentiation, JIT compilation, and parallelisation. It proceeds in the following stages :

1. **Model construction**
   - Input is a Calabi–Yau geometry (from databases like Kreuzer–Skarke or CICY) plus orientifold and flux choices.
   - Tools such as CYTools are used to build the effective data set (intersection numbers, periods, etc.).
2. **EFT module (Effective Field Theory in JAX)**
   - Constructs the 4D supergravity data: prepotential, Kähler potential, and flux superpotential.
   - Computes scalar potential, F-term conditions, period vectors, and Hessians via JAX auto-differentiation.
   - JIT and vectorisation (vmap) make this scalable to O(100) moduli.
3. **Optimisation module**
   - Translates extremisation (F-flatness, DIW = 0) into a numerical optimisation/root-finding problem.
   - Uses scipy.optimize.root as default solver (with potential future use of GPU-based or gradient descent optimisers).
4. **Sampling module**
   - Chooses fluxes and initial guesses for moduli values.
   - Employs methods such as ISD-based sampling (using the imaginary self-dual condition) to generate promising starting points.
   - This step is crucial for efficiency and success rate.
5. **Filter module**
   - Ensures identified solutions are trustworthy vacua.
   - Performs checks such as positivity of the Hessian, consistency with tadpole bounds, and EFT control (e.g. large complex structure regime).

🔄 **Pipeline flow:**

UV input (geometry + fluxes) → EFT module builds scalar potential → Optimisation module searches for extrema → Sampling provides flux/initial guesses → Filter checks validity → Valid string vacua dataset.

This workflow enables JAXVacua to construct ~10^6 flux vacua per geometry even with dozens of complex structure moduli .

Do you want me to also make you a **flowchart-style diagram** of this workflow (e.g. boxes for each module with arrows), so you have a visual summary?

- # Can you give me a technical summary of the string theory constructions used in this paper (2107.04039)? This is for experts.

Here's a compact, expert-level summary of the **string theory constructions** used in the paper, focusing only on the UV setups and equations that define the flux vacua being searched.

Common IIB flux-vacuum framework

- **Background:** Type IIB on a Calabi–Yau threefold M with axio-dilaton $\phi=C_0+ie^{-\varphi}$ and complex-structure moduli z^a. Choose a symplectic 3-cycle basis $\{A_a,B^a\}$ with periods $\Pi(z)=(G_a,z^a)$ and $\Omega=z^a\alpha_a-G_a\beta^a$. Kähler potential
$K=-\log\!\left(-i\,\Pi^\dagger\Sigma\Pi\right)-\log\!\left(-i(\phi-\bar\phi)\right)$.

Turn on quantized RR/NSNS fluxes F_3,H_3 (integer vectors f,h in the symplectic basis). Superpotential and no-scale scalar potential:
$W=(f-\phi h)\!\cdot\!\Pi(z),\qquad V=e^K\!\left(K^{a\bar b}D_aW\,\overline{D_bW}+K^{\phi\bar\phi}D_\phi W\,\overline{D_\phi W}\right)$.

ISD/Minkowski vacua are imposed by the F-term conditions
$D_\phi W=\frac{1}{\bar\phi-\phi}\,(f-\bar\phi h)\!\cdot\!\Pi=0,\qquad$
$D_aW=(f-\phi h)\!\cdot\!(\partial_a\Pi+\Pi\,\partial_aK)=0$.

Flux-induced D3 charge $N_{\text{flux}}=\int F_3\wedge H_3=f\cdot\Sigma\cdot h$ is bounded by a model-dependent tadpole $0<N_{\text{flux}}<L_{\max}$. An $SL(2,\mathbb Z)$ redundancy is fixed by restricting ϕ to $|\mathrm{Re}\,\phi|<\tfrac12,\ |\phi|>1$.

Background A: Conifold region of $WP^4_{1,1,1,1,4}$

- **Geometry & orientifold:** Hypersurface $\sum_{i=1}^{4}x_i^8+4x_0^2+8\psi\,x_0x_1x_2x_3x_4=0$ with $(h^{1,1},h^{2,1})=(1,149)$. Consider the orientifold $x_0\to -x_0,\ \psi\to -\psi$ descending from an F-theory fourfold $X_A=WP^5_{1,1,1,1,8,12}$. The Euler characteristic gives $L_{\max}=\chi(X_A)/24=23328/24=972$.
- **Truncation & periods:** A discrete symmetry $\Gamma=\mathbb Z_8^2\times \mathbb Z_2$ leaves only ψ uncharged, so one keeps ϕ and a single complex-structure modulus. Near the conifold, set $x=1-\psi$ and expand
$\{G_1,G_2,z_1,z_2\}(x) \text{ with the standard } \ln x \text{ branch in } G_2$,
using numerically given constants $a_0,a_1,\ldots,d_1$. The flux vector is $N=(f_1,f_2,f_3,f_4,h_1,h_2,h_3,h_4)$.
- **Analytic ISD solution near the conifold:** Solving F-terms yields, to leading order,
$\phi=\frac{f_1\bar a_0+f_2\bar b_0+f_3\bar c_0}{h_1\bar a_0+h_2\bar b_0+h_3\bar c_0}+\mathcal O(|x|\ln|x|)$,
and an implicit equation for $\ln x$ in terms of fluxes and the period data (via μ_0,μ_1). These control the approach to the conifold locus and allow explicit evaluation of W_0 and g_s.
- **Targets used in searches:** For benchmarking, the paper fixes a large $|W_0|=50{,}000$ and enforces the gauge/tadpole constraints $0<N_{\text{flux}}<972$ and ϕ in the $SL(2,\mathbb Z)$ fundamental domain.

Background B: Symmetric six-torus T^6 (one-modulus truncation)

- **Setup & reduction:** Start with T^6 parameterized so that $\Omega=dz_1\wedge dz_2\wedge dz_3$ with $dz_i=dx_i+\tau^{ij}dy_j$. In the **symmetric** truncation $\tau^{ij}=\tau\delta^{ij}$, the moduli are $\{\tau,\phi\}$ and the fluxes reduce to eight integers $(a_0,a,b,b_0;c_0,c,d,d_0)$ via
$F_3=a_0\alpha_0+a\,\alpha_{ii}+b\,\beta_{ii}+b_0\beta_0,\quad$
$H_3=c_0\alpha_0+c\,\alpha_{ii}+d\,\beta_{ii}+d_0\beta_0$.

The superpotential becomes
$W=P_1(\tau)-\phi\,P_2(\tau),\quad$
$P_1=a_0\tau^3-3a\tau^2-3b\tau-b_0,\quad$
$P_2=c_0\tau^3-3c\tau^2-3d\tau-d_0$.

F-terms reduce to $P_1(\tau)-\bar\phi P_2(\tau)=0$ and $(P_1-\phi P_2)=(\tau-\bar\tau)(P_1'-\phi P_2')$. Eliminating $\phi=P_1/P_2$ yields a cubic equation for $x=\mathrm{Re}\,\tau$ (with auxiliary polynomials q_i and coefficients α_i); then $y=\mathrm{Im}\,\tau$ follows from two real equations.
- **Flux D3 charge and tadpole:**
$N_{\text{flux}} = b_0 c_0 - a_0 d_0 + 3(bc-ad)$.

For the standard orientifold $T^6/\mathbb Z_2$ with 64 O3-planes one uses $L_{\max}=16$ and imposes $0<N_{\text{flux}}<16$ together with the ϕ fundamental-domain cut.
- **Targets used in searches:** Two tasks are considered: (i) **small superpotential** $|W_0|<10$; (ii) **weak coupling** $g_s\simeq 0.3$ (i.e. $2.5<\mathrm{Im}\,\phi<4$), both with the same gauge/tadpole constraints.

What is actually "varied" in the UV?

Across both backgrounds the paper varies the **integer flux vector**
$N=(f_1,\ldots,f_{b_3},h_1,\ldots,h_{b_3})$
subject to the tadpole bound, while **solving exactly** for $(\phi,\text{complex structure})$ via the analytic conifold expansions or the torus cubic. The **Kähler moduli** are spectators (no-scale), so the vacua are ISD Minkowski solutions of the complex-structure/axio-dilaton subsystem. The $SL(2,\mathbb Z)$ redundancy is fixed by the fundamental-domain cut to avoid overcounting.

If you want, I can now extract the explicit numerical period data and the μ_0,μ_1 definitions (conifold) or write out the cubic's coefficients α_i in terms of $(a_0,a,b,b_0;c_0,c,d,d_0)$ for quick implementation.

- Which ML framework does this paper (1810.07703) use and which cluster properties do they study?

I've gone through the paper you uploaded (Ntampaka et al., A Deep Learning Approach to Galaxy Cluster X-ray Masses, arXiv:1810.07703). Here's the breakdown:

Machine Learning framework

• They use a Convolutional Neural Network (CNN).

• Implementation: Keras with a TensorFlow backend .

•Architecture: three convolution + pooling layers for feature extraction, followed by three fully connected layers (with ReLU activations, dropout, and global average pooling).

•Optimizer: Adam, with mean squared error loss.

Cluster properties studied

•They focus on galaxy cluster masses inferred from mock Chandra X-ray observations.

•Training data: 7,896 mock X-ray images from 329 clusters in the IllustrisTNG TNG300 simulation .

•Target quantity: $M_{500}c$ (cluster mass within radius $R_{500}c$, the radius where the mean density is 500 times the critical density).

•They analyze how CNNs learn to ignore the cluster cores (inner ≲0.2 $R_{500}c$), which are noisy/scattery due to feedback physics, and instead use outskirts (surface brightness and morphology) to improve mass prediction.

•Performance: CNN achieves ~8–12% scatter in mass prediction, compared to 15–18% for traditional X-ray luminosity methods.

👉 So in short: CNN (Keras/TensorFlow) trained on mock Chandra X-ray images of IllustrisTNG clusters to predict cluster mass ($M_{500}c$), with analysis of sensitivity to core vs outskirts properties.

Would you like me to also check whether they study secondary cluster properties (like dynamical state, morphology, or gas clumping) explicitly, or is your interest mainly in the primary ML target (mass)?

Reviewing and improving text

- Can you give me a technical summary of the formalism for flux vacua used in this paper?
- Reviewer: Make sure …
- Based on the reviewer comments, can you improve your summary.

```
review_prompt = f"""
Given the context:
{joined_context}

And the draft paragraph:
{paragraph}

Evaluate whether it satisfies the prompt: '{prompt}'.
Provide clear feedback on clarity, correctness, and relevance.
"""
```

```
summary_prompt = f"""
Given the following reviews:
{chr(10).join(reviews)}

Summarize the common points. If most reviewers prefer this version over a prior one
(if available), state so clearly.
"""
```

```
improve_prompt = f"""
Prompt: {prompt}

Context:
{joined_context}

Current version:
{old_paragraph}

Review summary:
{review_summary}

Please revise the paragraph to address the review suggestions.
"""
```

Prompt: Provide a short review on how supersymmetric flux vacua for complex structure moduli are found. [gpt4o]

Supersymmetric flux vacua for complex structure moduli in string theory are typically found by solving the F-term conditions derived from the superpotential and Kähler potential of the effective four-dimensional theory. In the context of Type IIB string theory compactified on Calabi-Yau (CY) orientifolds, the superpotential is given by the Gukov-Vafa-Witten (GVW) form, $W = \int G_3 \wedge \Omega$, where G_3 is the complexified three-form flux and Ω is the holomorphic three-form of the CY manifold. The Kähler potential for the complex structure moduli Z^i and the axio-dilaton τ is $K = -\log(-i(\tau - \bar{\tau})) - \log(-i\int \Omega \wedge \bar{\Omega})$. Supersymmetric vacua are found by imposing the conditions $D_I W = \partial_I W + (\partial_I K) W = 0$ for all moduli fields I, which ensure that the scalar potential derived from the supergravity action is minimized. These conditions imply that the flux G_3 must be imaginary self-dual (ISD), i.e., $\star_6 G_3 = i G_3$ in terms of the Hodge star operator on the CY manifold. The solutions to these equations are constrained by tadpole cancellation conditions, which require the total D3-brane charge induced by the fluxes to be balanced by the presence of orientifold planes and D-branes. Finding explicit solutions involves a combination of analytical techniques, such as solving Picard-Fuchs equations for periods and using mirror symmetry, and numerical methods, including root-finding algorithms and sampling of flux configurations, to explore the vast landscape of possible vacua.

Supersymmetric flux vacua for complex structure moduli in Type IIB string theory are identified by solving the F-flatness conditions, which require that the derivatives of the superpotential W with respect to the moduli vanish, i.e., $D_I W = 0$. This process involves compactifying the theory on a Calabi-Yau manifold—a complex, multi-dimensional shape that preserves some supersymmetry—and introducing 3-form fluxes. These fluxes generate a scalar potential for the moduli fields, which are parameters describing the shape and size of the extra dimensions. The F-flatness conditions ensure that the vacuum preserves supersymmetry by finding critical points of the scalar potential where its first derivatives vanish.

These conditions are typically addressed in the large complex structure regime, where the prepotential governing the moduli space is well-understood, allowing for computational control. Solutions to these equations correspond to points in moduli space where the scalar potential is minimized. These points are characterized by specific values of the moduli that satisfy both the F-flatness and the tadpole cancellation conditions, the latter ensuring that the total charge from the fluxes and other sources is balanced.

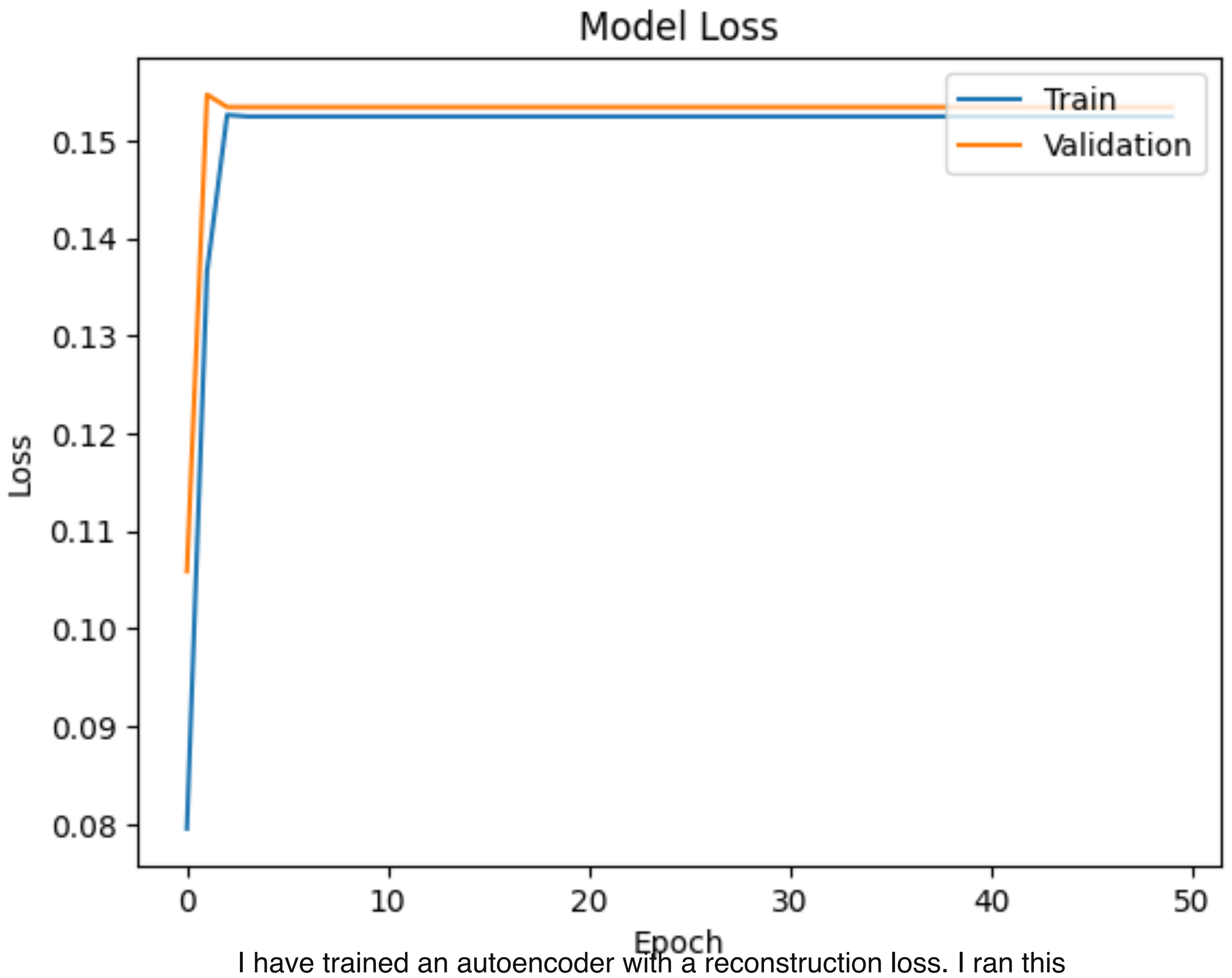
Recent advances in numerical methods, such as automatic differentiation and just-in-time compilation, have significantly improved the efficiency of finding these solutions. Automatic differentiation allows for precise calculation of derivatives, which is crucial for solving the F-flatness conditions, while just-in-time compilation accelerates the computational process. These techniques enable the exploration of large samples of vacua across different geometries, which is crucial for understanding the landscape of possible low-energy physical theories derived from string theory. These vacua are particularly interesting because they influence low-energy properties like the gravitino mass—a parameter related to supersymmetry breaking—and the effective four-dimensional cosmological constant, which are important for connecting string theory to observable physics.

For readers less familiar with the topic, it's helpful to note that the Calabi-Yau manifold is a key structure in string theory that allows for compactification while preserving some supersymmetry. The tadpole cancellation condition is a requirement that ensures the consistency of the theory by balancing charges in the compactified dimensions. The use of advanced numerical techniques helps overcome the computational challenges involved in solving the complex equations that describe these systems, enabling researchers to probe deeper into the theoretical landscape and its potential implications for understanding the fundamental nature of the universe.

Coding, running and debugging

- Write me some code …
- Writing files LLM output -> file.py; Running files subprocess.run(file.py)
- When running the code, I get the following error. Please provide a new version of the code to avoid this error.
- Over to the example with gpt-4o and local LLM (some tasks are cheap!)

Analyse results



I have trained an autoencoder with a reconstruction loss. I ran this code:

```
# Compile the autoencoder
autoencoder.compile(optimizer=Adam(), loss=MeanSquaredError())

# Train the autoencoder
history = autoencoder.fit(x_train, x_train,
                epochs=50,
                batch_size=256,
                shuffle=True,
                validation_data=(x_test, x_test))

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'], loc='upper right')
plt.show()
```

It produced the following plot.

Please comment on what you see and what do you think about these results in terms of the quality of this autoencoder.

Looking at your training/validation loss plot:

1. **Loss values**
   - Training starts around ~0.08 and quickly jumps to ~0.15 after the first epoch.
   - Both training and validation losses then remain essentially flat at ~0.152 for the rest of training.
   - There is almost no improvement beyond the first few epochs.
2. **Training vs Validation**
   - Training and validation losses are almost identical → this indicates no overfitting (the model generalizes equally well on train and test).
   - But it also shows the model is **not improving further** — it seems stuck at a plateau very early.
3. **Interpretation for autoencoder quality**
   - The fact that both losses stabilize so quickly suggests the autoencoder has **limited capacity** (too simple to reduce the reconstruction error further), or the **learning rate is too high**, preventing finer optimization.
   - The reconstruction error (~0.15 MSE) might be relatively large, depending on your input data scale (e.g., if pixel values are normalized between 0 and 1, then this is a noticeable reconstruction error).
   - Since loss stabilizes almost immediately, the model may not be learning meaningful latent representations — it could be just learning a shallow mapping.
4. **Next steps to improve**
   - **Check learning rate**: Try lowering the Adam learning rate (e.g., 1e-3 → 1e-4) to allow smoother convergence.
   - **Increase model capacity**: If your autoencoder is too small (few layers, few neurons), add more layers/filters.
   - **Regularization**: Even though you don't see overfitting now, adding dropout or weight decay can allow for more expressive models without overfitting.
   - **Normalization**: Ensure input data is normalized (e.g., scaled to [0,1] or standardized).
   - **Alternative loss functions**: If reconstructing images, try perceptual loss, SSIM, or BCE loss (for binary data) instead of plain MSE.

👉 In summary: The autoencoder is **not performing very well** — it converged quickly to a plateau without learning richer reconstructions. This usually means either the architecture is too simple for the task, or the optimization setup (learning rate/loss) is preventing progress.

Do you want me to suggest a set of concrete **architecture tweaks** (e.g. convolutional autoencoder setup) that usually work much better for image data?

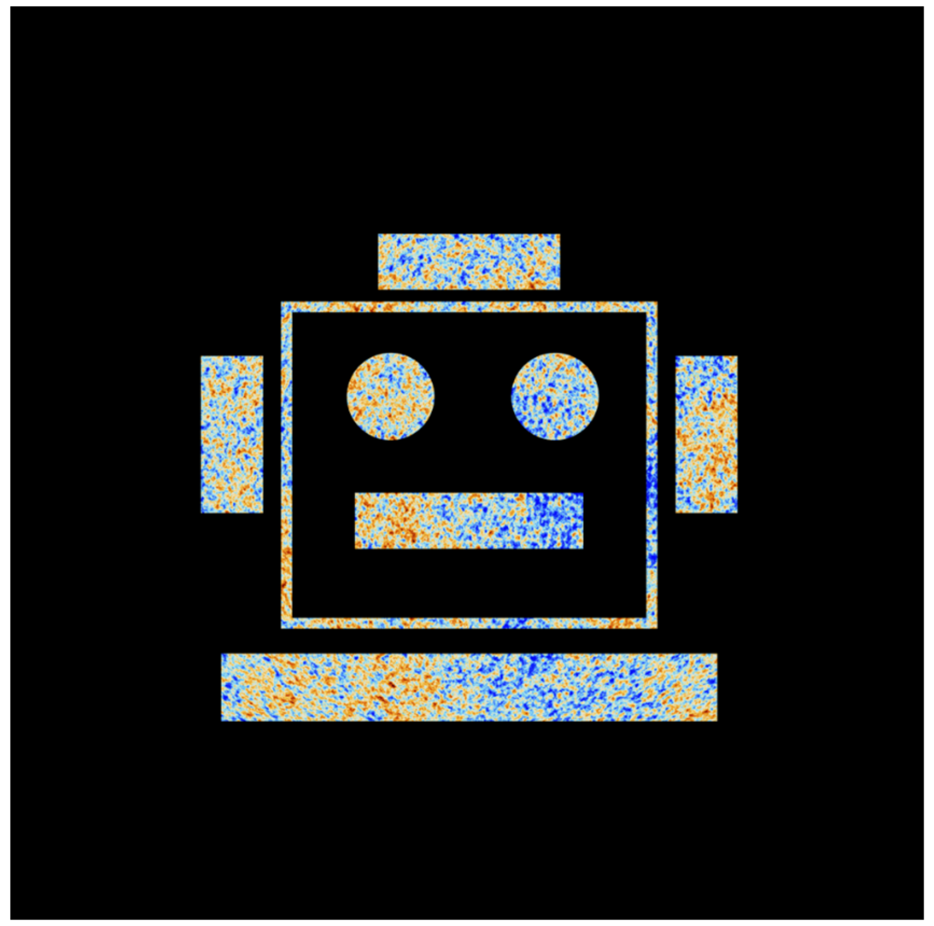
Breaking a prompt down in manageable sub-tasks

Keeping the LLM focused and not running out of context length

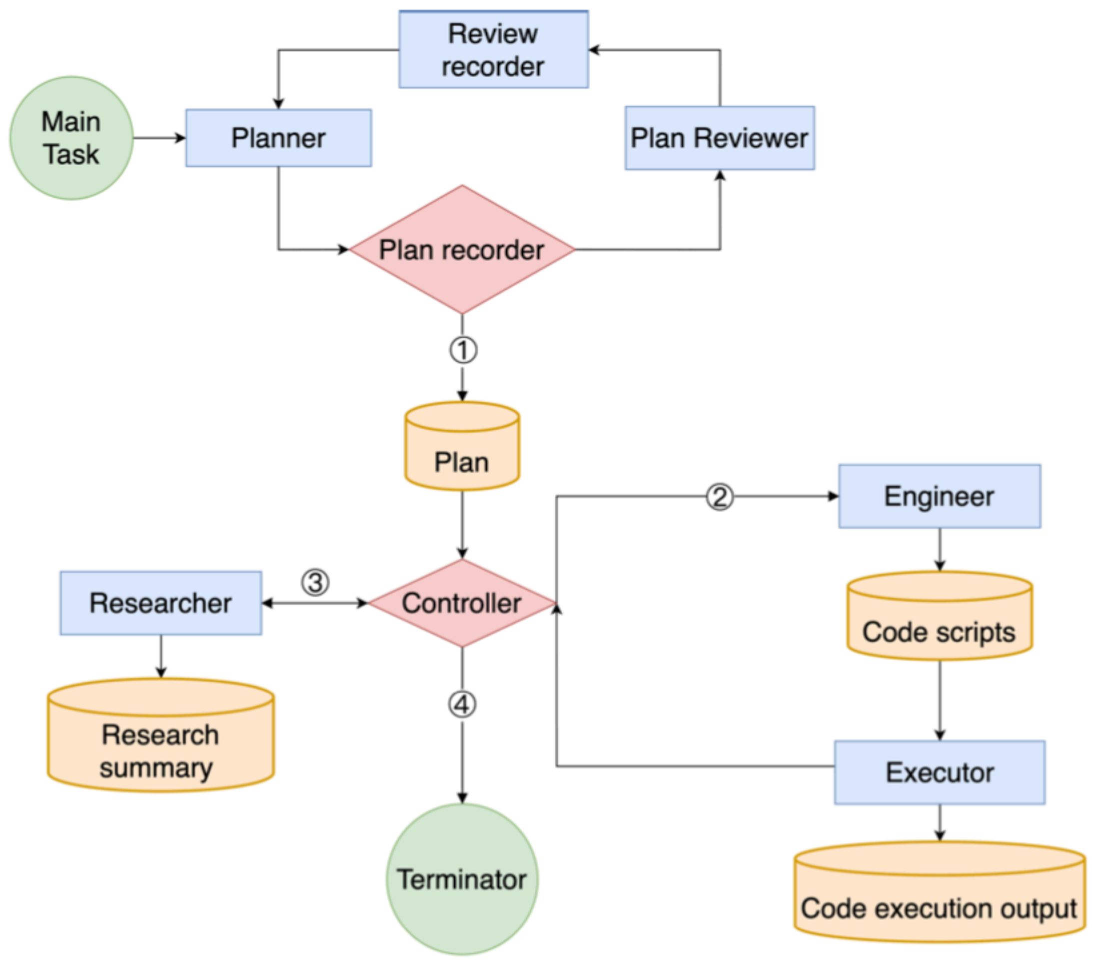
- **Make it possible with a planner:** You are a project planning agent. The user would like to address this task: {add user prompt}
  You should return a plan on how to address this task breaking the task into manageable steps.

- **For instance:** write code, execute code, write a summary about the results.

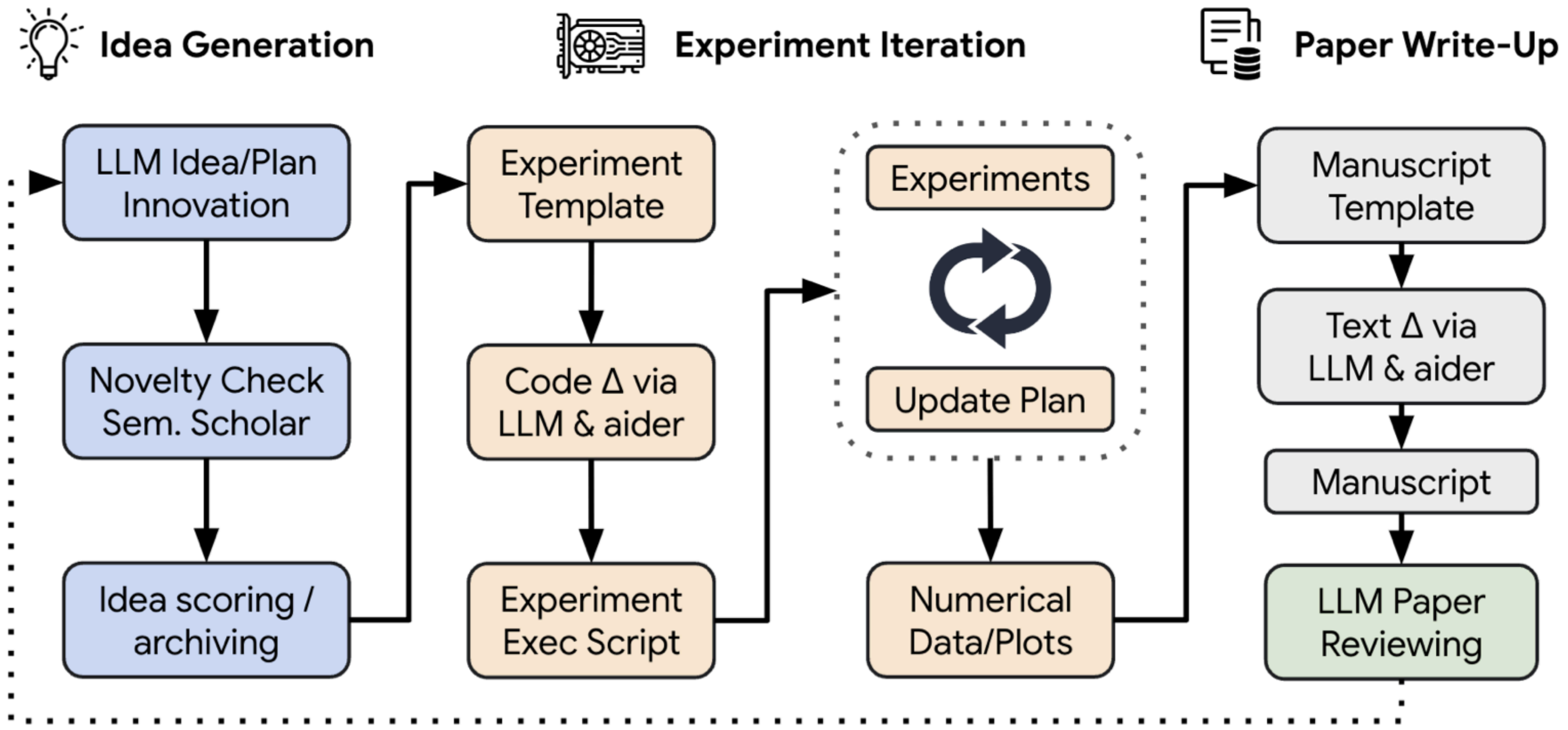
Public results/systems



CMBAGENT
multi-agent systems for autonomous discovery

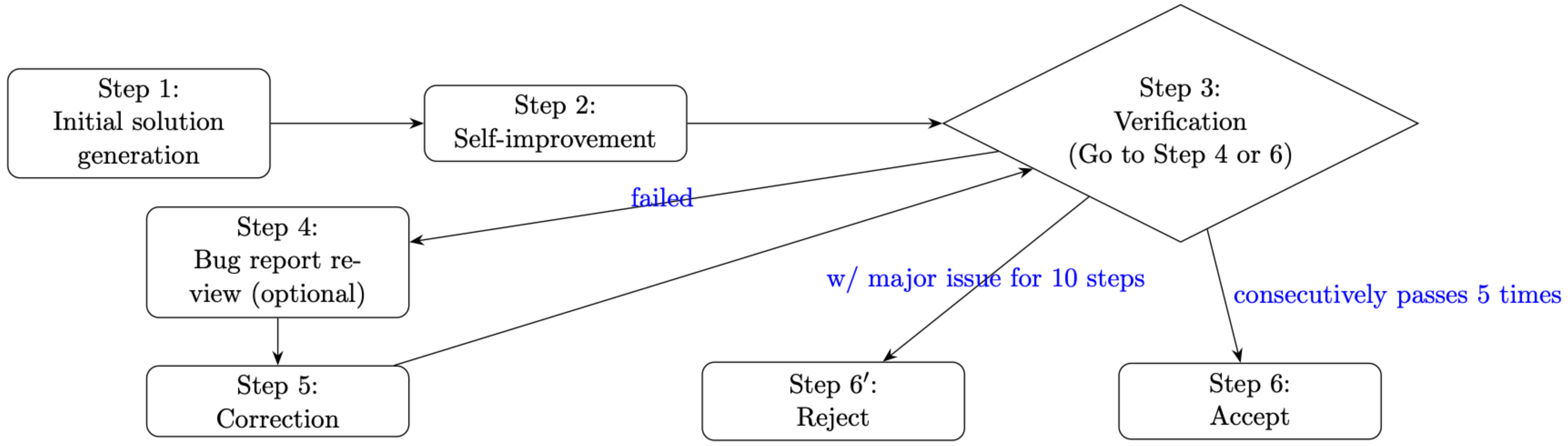


2507.07257



from AI Scientist

https://github.com/CMBAgents/cmbagent



2507.15855

Lu et al.: The AI Scientist 2408.06292
Laverick et al.: Multi-Agent System for Cosmological Parameter Analysis 2412.00431
Moss: The AI Cosmologist I: An agent system for automated data analysis 2504.03424
TPBench (Münchmeyer et al.): tpbench.org

IMO Math Olympiad (Huang, Yang): 2507.15855
…

The key questions ahead

- How good are they at taking a paper and writing the implementation of an algorithm described in the paper? [translation from text to code]
- Can we successfully enforce the use of methods from the appropriate literature rather than some incorrect and incomplete “one-shot” answers.
- Can they deal with the ups and downs of our standard research projects independently? [long-term stability without human in the loop]

Things not worth talking about right now…

- How about creating new ideas? Personally most of our ideas are straight-forward open questions that are mentioned in some papers, talks, asked as questions in talks.

Here are **five open research questions** that naturally emerge from the paper *“Deep observations of the Type IIB flux landscape”* (arXiv:2501.03984) and could form the basis for new projects:

1. **Extension to Non-Supersymmetric Vacua**

The current framework focuses on supersymmetric ISD flux vacua. How can the algorithm be generalized to systematically explore *non-supersymmetric* solutions, where the ISD condition no longer applies? This is crucial for understanding metastable de Sitter vacua and F-term uplifts .

2. **Mathematical Structures in Superpotential Distributions**

The observed distributions of the flux superpotential W_0 in the complex plane display arcs, voids, and symmetry-breaking features . What mathematical symmetries or geometric structures underlie these patterns? Are they generic to certain Calabi–Yau topologies or tied to specific flux hierarchies?

3. **Mass Hierarchies and Phenomenology**

The study reveals significant hierarchies and axion–saxion mixing in moduli masses . How do these hierarchies influence phenomenological properties such as supersymmetry-breaking scales, the cosmological moduli problem, or the viability of inflationary and de Sitter scenarios?

4. **Bridging Local and Global Landscape Properties**

Local deviations from statistical expectations in vacuum densities were found . Can one develop refined analytic tools that accurately predict vacuum densities across different regions of moduli space? This would help reconcile explicit constructions with global statistical landscape approaches.

5. **Scalability to Higher-Dimensional Moduli Spaces**

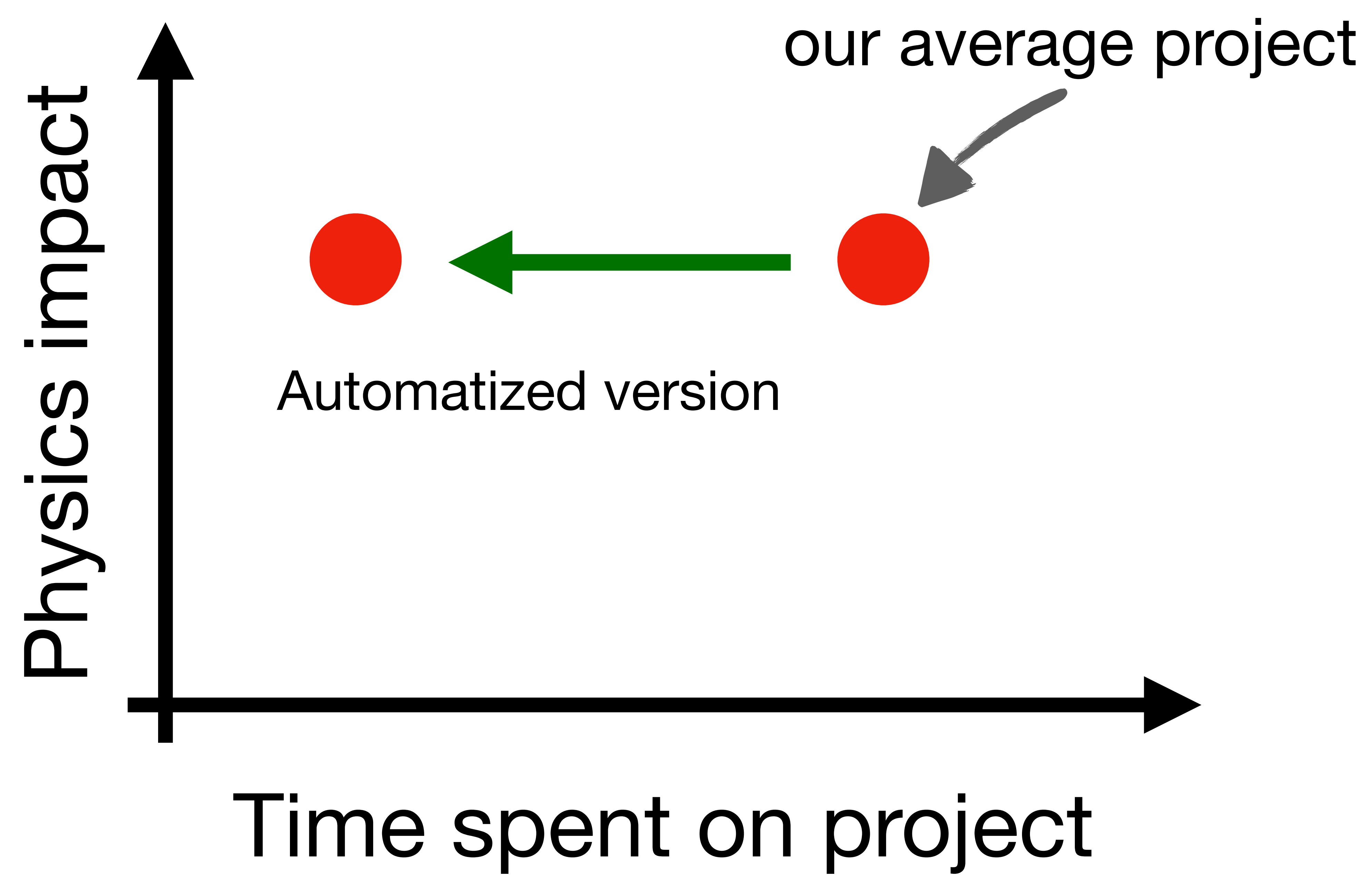
The paper focuses on a two-moduli model. How can the proposed methods be scaled to higher h^{2,1} Calabi–Yau spaces, such as those in the Kreuzer–Skarke database, while retaining computational feasibility ? This is necessary for testing conjectures (e.g. tadpole bounds, rarity of small W_0) at large h^{2,1}.

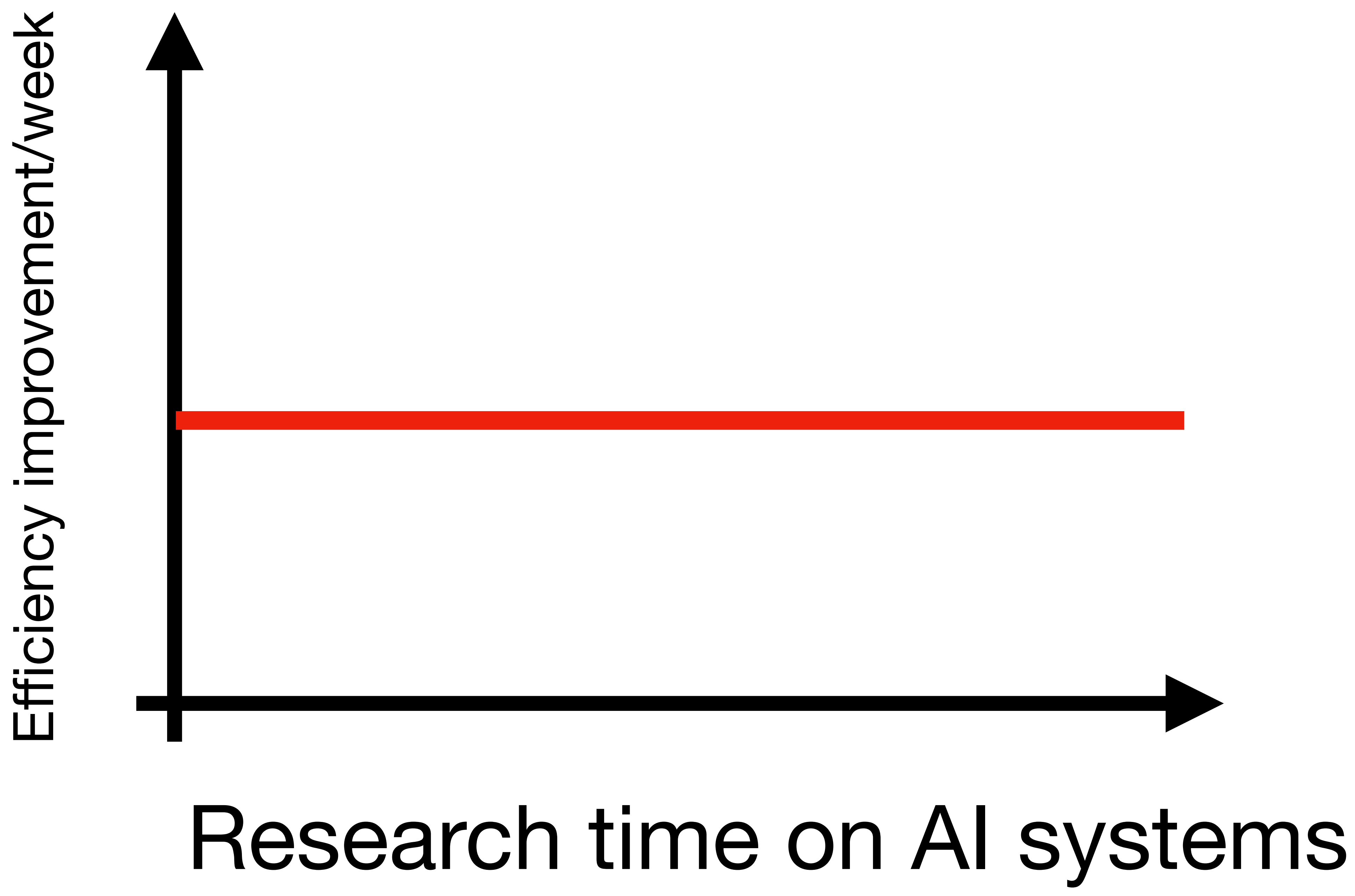
Would you like me to **reframe these questions into concrete project proposals** (e.g. with objectives, methods, expected outcomes), or keep them as high-level open directions?

Quo vadis?

Personal thoughts

- We can significantly accelerate most research projects.
- The technology is around and we need to put it together in efficient researcher systems.
- In one year we are likely to have systems that can do ~90% of current research projects with vanishingly small human time per project.
- I expect that the collective power of many “small” papers leads to significant scientific contributions! Happy to bet on this.
- Very exciting times ahead…
- Important to attempt apples to apples comparisons with human and machine researchers.





Thank you!

Happy 60th Birthday Steve!